U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A024 253

COMPUTER PROGRAMS FOR THE AUSEX (AIRCRAFT UNDERSEA SOUND EXPERIMENT) AIR-WATER ACOUSTIC PROPAGATION MODEL

BOLT BERANEK AND NEWMAN, INCORPORATED

PREPARED FOR
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

28 JANUARY 1976

AD A024253

BOLT BERANEK AND NEWMAN INC

135152

TECHNICAL MEMORANDUM
W307

COMPUTER PROGRAMS FOR THE AUSEX AIR-WATER ACOUSTIC PROPAGATION MODEL

D. Sachs
L. Sledjeski
R. Stern

Contract N00014-75-C-0532
ARPA Order 2909, MOD #2, 7/7/75
BBN Job 10054
BBN Technical Memorandum W307

28 January 1976

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0532.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: LCDR Wesley Jordan
Tactical Technology Office

Prepared by:

Bolt Beranek and Newman Inc.
1701 North Fort Myer Drive
Arlington, Virginia 22209

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) BOLT BERANEK AND NEWMAN INC.	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

Computer Programs for the AUSEX Air-Water Acoustic Propagation Model

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Memorandum W307

5. AUTHOR(S) (First name, middle initial, last name)

David Sachs
Leo Sledjeski
Raya Stern

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
28 JAN 1976	147	5

8a. CONTRACT OR GRANT NO. N00014-75-C-0532	9a. ORIGINATOR'S REPORT NUMBER(S) TM W307
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

UNLIMITED

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Defense Advanced Research Projects Agency

13. ABSTRACT

This report is a user's guide for a computerized mathematical model development in support of the Aircraft Undersea Sound Experiment (AUSEX) Program. The AUSEX Program is funded and directed by the Defense Advanced Research Projects Agency (DARPA).

The subject mathematical model is termed the AUSEX Acoustic Propagation Model and it describes the acoustic propagation from a moving sound source in air, to and across a rough air-water interface, and subsequently through the water to an arbitrarily located point-acoustic receiver.

This report describes the computer program architecture and the input and output data associated with the program's use. Two examples of the program's application are included. In addition, the complete program listing for the Fortran IV coding is included.

A description of the mathematical basis for the Acoustic Propagation Model is contained in a companion report, BBN TM W311, "AUSEX Air-Water Acoustic Program Model."

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Acoustic Propagation Model						
Air-Water Interface						
Computer Program						

COMPUTER PROGRAMS FOR THE AUSEX AIR-WATER ACOUSTIC PROPAGATION MODEL

D. Sachs
L. Sledjeski
R. Stern

Contract N00014-75-C-0532
ARPA Order 2909, MOD #2, 7/7/75
BBN Job 10054
BBN Technical Memorandum W307

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0532.

28 January 1976

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: LCDR Wesley Jordan
Tactical Technology Office

Prepared by:

Bolt Beranek and Newman Inc.
1701 North Fort Myer Drive
Arlington, Virginia 22209

16

ABSTRACT

This report is a user's guide for a computerized mathematical model development in support of the Aircraft Undersea Sound Experiment (AUSEX) Program. The AUSEX Program is funded and directed by the Defense Advanced Research Projects Agency (DARPA).

The subject mathematical model is termed the AUSEX Acoustic Propagation Model and it describes the acoustic propagation from a moving sound source in air, to and across a rough air-water interface, and subsequently through the water to an arbitrarily located point-acoustic receiver.

This report describes the computer program architecture and the input and output data associated with the program's use. Two examples of the program's application are included. In addition, the complete program listing for the Fortron IV coding is included.

A description of the mathematical basis for the Acoustic Propagation Model is contained in a companion report, BBN TM W311, "AUSEX Air-Water Acoustic Program Model".

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 Overview and Objective of the Report

The Defense Advanced Research Projects Agency (DARPA) has been sponsoring a research and development program termed AUSEX (Aircraft Undersea Sound Experiments). Briefly, the objective of the AUSEX Program is the development of generic algorithms for the detection, classification, and tracking of air vehicles by their underwater acoustic signatures.

The principal elements and their functional relationship are indicated in Fig. 1-1. One of the primary inputs to the detection, classification, tracking algorithm design is the outputs from the air-water acoustic propagation model.[1*] This model was designed to provide the algorithm designer with a detailed description of the signal field at a point in the ocean as a function of time, the signal being the radiated acoustic energy associated with an air vehicle (i.e. fixed and rotary wing aircraft and cruise missiles).

Other AUSEX Program elements have been concerned with the development of detailed acoustic source characterizations of air vehicles of interest.[2] And another dealt with the quantification of the effects of real ocean surfaces on the propagation of sound from air to water.[3]

The air-water propagation model was computer programmed as a tool for the detection algorithm designers. This technical memorandum is intended to serve as a user's guide and as such

*Superscript numbers identify references as listed on page R-1.

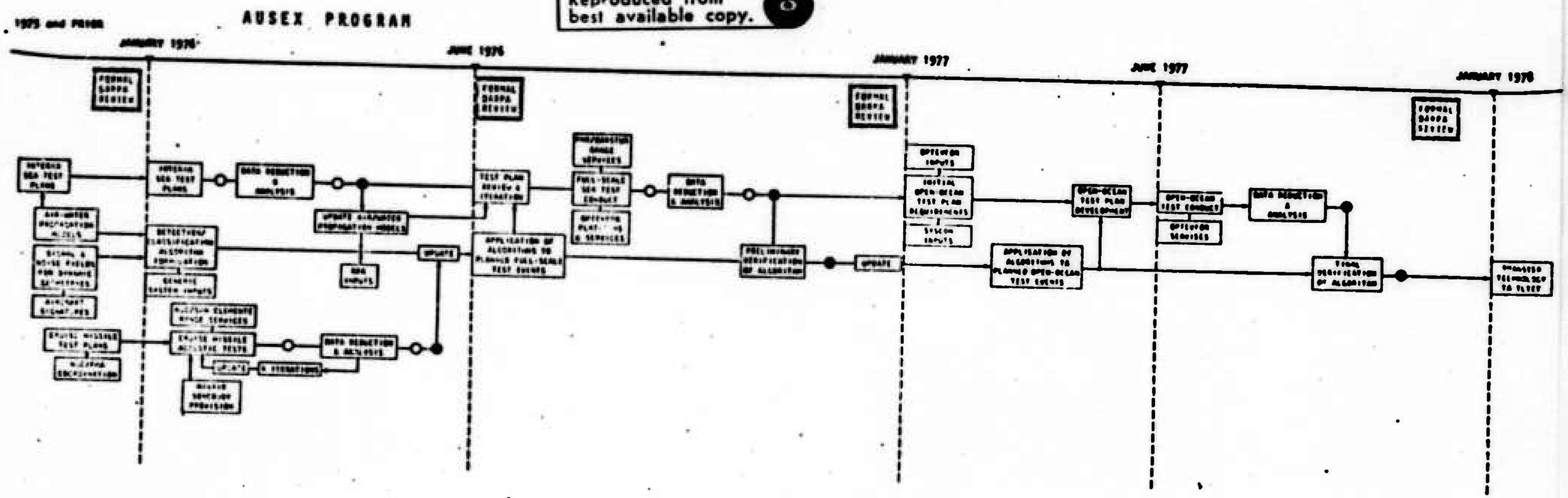


FIGURE 1-1 PRINCIPAL ELEMENTS OF THE AUSEX PROGRAM

it describes the computer program, its required input, and subsequent output. Typical examples of its utilization are given. The program listing in Fortran IV for the BBN TENEX System is also included. A description of the mathematical basis for the Propagation Model is given in BBN Technical Memorandum TM W311, "AUSEX Air-Water Program Model". This report is in publication.

1.2 Report Organization

Section 2 is a summary of the overall air-water acoustic propagation program. The program is designed as an executive with several subroutines which concern such things as the air-water interface model, the several distinct underwater acoustic propagation modes, the atmospheric acoustic propagation, and the dynamics of the air-source underwater-receiver encounter.

Section 3 displays some typical results in manually plotted form. In addition, this section includes some discussion of the individual results for the example cases chosen.

SECTION 2
PROGRAM SUMMARY

2.1 Intended Utilization

The ultimate objective of the AUSEX Program is the development of generic algorithms for the detection, classification, and tracking of air vehicles by their underwater acoustic signatures. The AUSEX air-water acoustic propagation model is intended to provide the detection algorithm designer with a detailed, time dependent description of the acoustic field produced at a (moving) point in the ocean by an air vehicle. The model output provides a primary information base which the designer will utilize to construct and optimize detection/classification/tracking algorithm schemes.

The user specifies atmospheric and ocean environmental parameters and air vehicle and receiver track parameters. The model code then marches through time as the encounter unfolds, calculating time histories of the following quantities:

- Transmission loss from air vehicle to the receiver for each underwater mode of propagation (i.e., direct path, bottom bounce, surface duct and convergence zone).
- Depression/elevation and azimuthal arrival angles for each propagation mode.
- Received frequency by propagation mode.
- Range variables.

In obtaining these variables, the model accounts for atmospheric propagation, air-water interface transmission, underwater propagation and source/receiver geometry dynamics.

The parameter space covered by the model is given in Table 2-1.

The family of model outputs for anticipated scenarios provides a hypothesis space for the algorithm designer to use in the synthesis of detection schemes.

2.2 Program Architecture

Figure 2-1 shows the basic program structure. There are two independent programs, one of which calculates direct path and bottom bounce outputs, while the other computes convergence zone and surface duct quantities.

For each of the programs, the input data is inserted and the program begins at a user-specified initial time and calls upon each of a pair of propagation subroutines, in turn. The subroutines are almost completely self-contained and mutually independent. (The subroutines are preceded by a sub-program which calculates a small number of common quantities.) Each of the propagation subroutines calculates all the pertinent output for only one type of underwater propagation mode. Incorporated in the subroutines are atmospheric propagation, air-water interface transmission, underwater propagation and source-receiver geometry dynamics. For a given time instant, each of the subroutines outputs time variables (emission time, arrival time), transmission loss, depression/elevation and azimuthal arrival angles, received frequency and range variables (range at emission time, range at arrival time). The time is then increased by a user-specified increment and the calculation process repeated until the incremented time variable exceeds a user-specified final time.

TABLE 2-1

PROGRAM PARAMETER SPACE

Air source parameters

Character:	point, omnidirectional
Speed:	any speed not large compared to the sound speed in air
Heights:	any
Lateral offsets:	> 0 yards
Radiation frequency:	> 0 - 10,000 Hz
Track:	straight line motion at constant height in any direction

Receiver parameters

Character:	point, omnidirectional
Speed:	small compared with water sound speed
Depths:	any small fraction of ocean depth (for instance, $\leq$ 600 ft)
Track:	straight line motion at constant depth

Range interval

Closest approach to inner edge of second convergence zone.

Sea state

Character:	fully developed or smooth (partially arisen seas and swells not included)
Wind speed:	any
Wind direction:	any

Table 2-1 continued.

<u>Atmosphere</u>	
Air temperature:	-10°F to 90°F
<u>Ocean</u>	
Bottom:	flat
Bottom types:	LRAPP types 3 and 5[5]
Sound speed profiles:	any two or three linearly segmented profile with a depth excess approximating a deep ocean winter profile (three segments) or summer profile (two segments)
Depth:	deep ocean depths shallow depths out to horizontal ranges where triple bottom bounce dominates

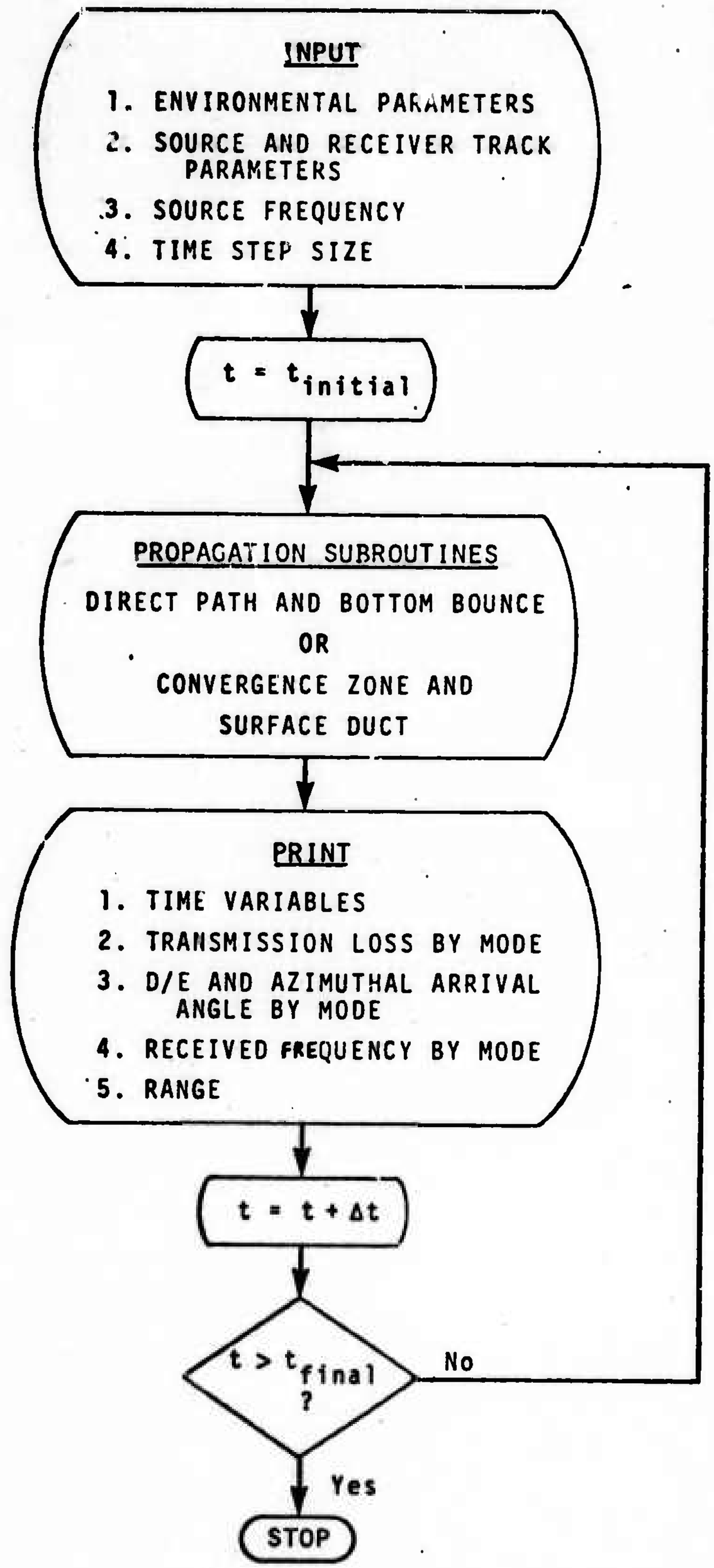


FIGURE 2-1 Schematic of Propagation Model Program

Initial and final time and time increments are selected by the user on the basis of the range interval to be covered and the temporal detail required.

No computer plotting capabilities are included in the current model code; all results must be hand plotted.

The program is written in FORTRAN IV. It is operational on the Interactive Sciences Corporation system and the BBN TENEX system.

Detailed descriptions of the component programs are given in Appendices A and B.

2.3 Input Data

The program accepts and inputs the quantities listed in Table 2-2. Refer to Figs. 2-2 and 2-3 for a pictorial definition of the variables.

The air source/receiver/time inputs define the encounter scenario and radiation frequency of the source. The atmospheric inputs quantify atmospheric propagation effects and the character of the air-water interface. Sound velocity profiles and bottom reflection losses are specified by the ocean medium input group.

Appendices A and B contain detailed descriptions of the input data format.

2.4 Output Data

Table 2-3 lists the major program outputs. See Fig. 2-2 for a schematic definition of some of the output variables.

For each value of absolute time, the output is grouped by propagation mode.

TABLE 2-2

PROPAGATION MODEL PROGRAM INPUTS

Times

starting time of encounter (in secs)

final time (in secs)

step size in time (in secs)

Air Source

v_{ax}, v_{ay}	—	cartesian components, in kts, of air source velocity
x_{ia}, y_{ia}	—	vector position of air source, in ft, at time t = 0 ("initial coordinates")
h	—	height, in ft, of air source
f	—	frequency of air source, in Hz

Receiver

D	—	depth (ft)
v_s	—	receiver speed along x-axis (in kts)
x_{is}	—	x-position of receiver (in ft) at time t = 0 ("initial coordinates")

Atmosphere

air temp.	—	either greater or less than 50°F
U	—	wind speed (kts)
n_{sx}, n_{sy}	—	cartesian direction cosines of wind velocity vector (dimensionless)

Ocean Medium

c_s	—	surface sound speed (ft/sec)
c_{min}, d_{min}	–	sound speed at sound speed profile minimum (ft/sec), depth of profile minimum (ft)

Table 2-2 continued.

c_b, D_B	—	sound speed at ocean bottom (ft/sec), depth to bottom (ft)
c_{max}, d_s	—	sound speed at bottom of surface duct (if one exists) (ft/sec), depth of surface duct (ft)
Bottom type	—	either LRAPP type 3 or 5

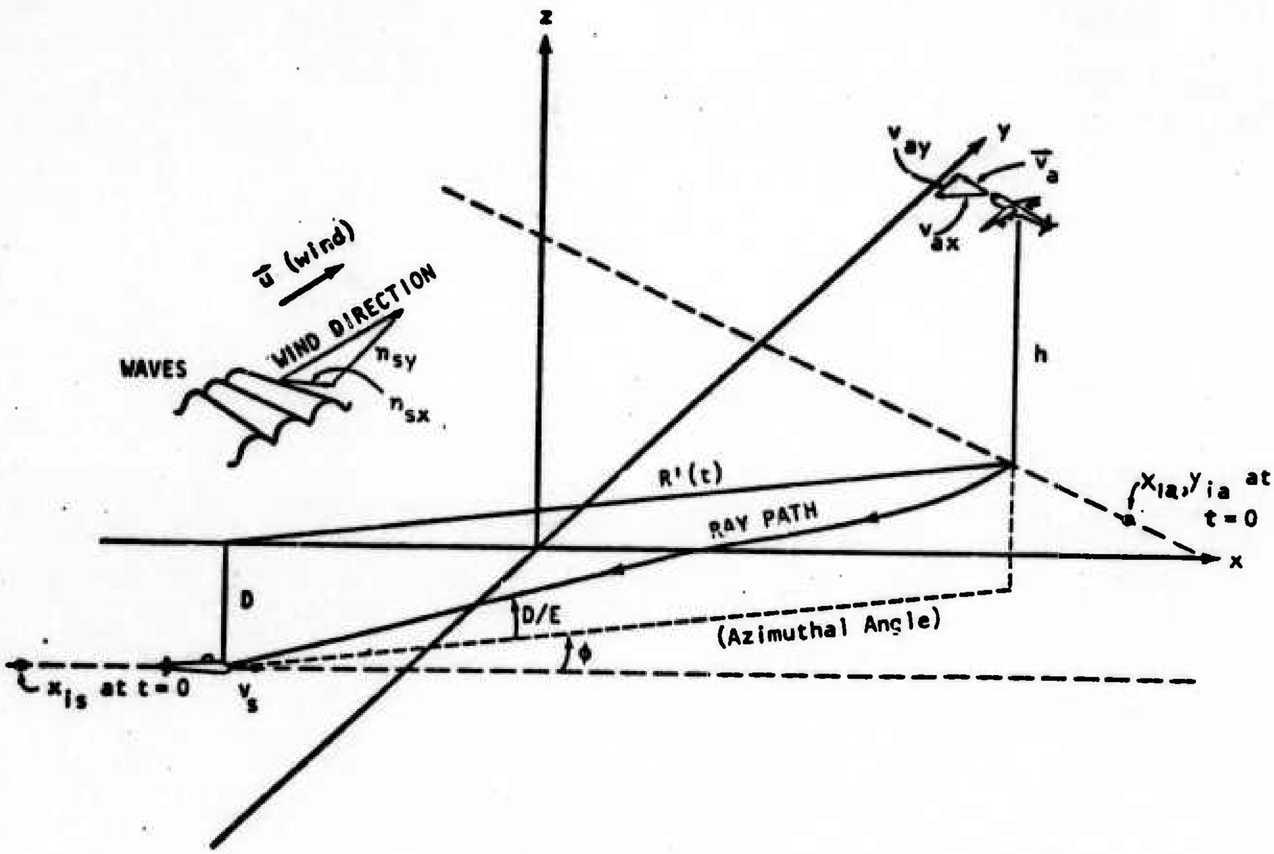
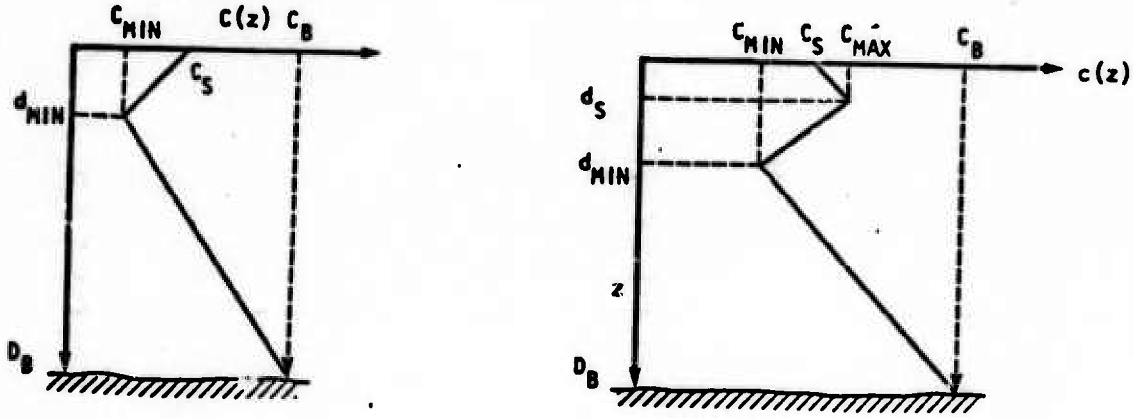


FIGURE 2-2   Schematic of Surface and Source / Receiver Variables.



(a) No surface duct

(b) Surface duct present

FIGURE 2-3 Schematic of Deep Ocean Sound Speed Profiles Permitted in Program. (a) is a Representative Summer Profile with no Surface Duct. (b) is a Representative Winter Profile with a Surface Duct. A Depth Excess is Always Assumed ($C_b > C_S$ in (a), $C_b > C_{MAX}$ in (b)). When a Surface Duct is Present, C_{MIN} is always Taken to be less than any Other Sound Speed.

TABLE 2-3

PROGRAM OUTPUTS

Propagation Mode Characterization

- Propagation mode type (i.e., single bottom bounce, direct path, etc.)

Time Variables

- absolute time

- arrival time when ray reaches receiver

- time difference between arrival time and time of arrival of direct ray from CPA point

Range Variables

- horizontal source/receiver range at time ray is emitted by air source (R'(t) in Fig. 2)

- horizontal source/receiver range at time ray is received

Arrival Angle Variables

- depression/elevation angle(s) of arriving ray(s) at arrival time of ray(s) for each propagation mode ("D/E" in Fig. 2)

- azimuthal arrival angle of arriving ray(s) at arrival time of ray(s) for each propagation mode (ϕ in Fig. 2)

Received Frequency

- ratio of received frequency to source frequency of arrival(s) at arrival time for each propagation mode

Table 2-3 continued.

Transmission Loss Variables

- transmission loss of received pressure level for each separate propagation mode, at appropriate arrival time, referred to air source level at 1 yd from the source, including water volumetric and bottom losses (where appropriate) and air volumetric losses, for both smooth and wind-driven ocean surfaces.

Following the mode identification, a number of time variables are given. The first is the time when the source radiates. Since the source is assumed to be continuously radiating, this time is the same as the absolute time. The arrival time when the ray reaches the receiver is given next. Finally, the time difference between the arrival time and an arbitrarily defined reference time is given. This reference time, which only has physical significance if the distance of closest approach is less than several kiloyards, is the time of arrival of the direct path ray. The program calculates direct path information as though the ocean were isospeed, regardless of the actual sound velocity profile. Physically, this is a good approximation for source/receiver ranges of several kiloyards or less. For greater ranges, a direct path connecting the effective surface source and the receiver will not exist because of refraction. The program, however, will still calculate a direct path arrival time. The user is consequently advised to ignore the time difference output variable when the encounter CPA is more than one or two kiloyards.

Two range variables are calculated. The first is the horizontal source/receiver range at the time the source radiates (same as absolute time). The second is the horizontal source/receiver range when the signal is received (i.e., at the arrival time).

Angle information includes the depression/elevation and azimuthal arrival angle(s) at the appropriate arrival time.

The ratio of received frequency to source frequency (Doppler shift) for each arrival is given.

The transmission loss for each arrival is calculated for both a smooth and wind-driven air/water interface, at a single source frequency. The transmission loss is defined as 10 $\log_{10}$ of the ratio of the mean squared received pressure to the square of the air source pressure at one yard from the source.

Broadband noise transmission loss may be treated by subdividing the noise spectrum into a number of frequency bands and using the single frequency results as an estimate of the transmission loss for each sub-band.

SECTION 3

TYPICAL RESULTS

Sample outputs of the programmed algorithm are presented and discussed in this section.

3.1 Representative North Atlantic Environmental Conditions

The input parameters selected for the first example are summarized in Table 3-1.

The sound speed profiles are chosen to be representative of North Atlantic mid-latitude profiles, and represent a compromise between the FNWC profiles of area 50[4] and the profiles for Marsden square 078, as given in the LRAPP volumes.[5*] The bottom type is a low loss type which occurs frequently in the North Atlantic area.

Sea State 3 (average wind speed 8.8 kts) occurs with 44% probability in the summer and 51% probability in the winter in Marsden square 078 and is taken to be representative of the area's general wind conditions. There is no reported predominant wind direction; a direction of 45° with respect to the air source track is assumed arbitrarily and should provide sufficient generality.

Air temperatures are taken to be less than 50°F for the winter profile and greater than 50°F for the summer profile. (The program actually uses air loss curves representative of 0°F for the case when the temperature is less than 50°F; when the air temperature is greater than 50°F, it uses air loss curves representative of 70°F.)

---

* Fleet Numerical Weather Central (FNWC) and Long Range Acoustic Propagation Project (LRAPP) have surveyed the deep ocean archival data and established representative sound speed profiles and bottom loss curves.

TABLE 3-1

EXAMPLE INPUT PARAMETERS

SOURCE/RECEIVER

Air Source:

Speed	= 220 kts
Height	= 10,000 ft
Frequency	= 150 Hz

Receiver:

Speed	= 7 kts
Depth	= 400 ft

Source/Receiver Tracks:

Track directions: parallel

Closest point of approach: 4 kyds

Range: extends from CPA to beyond first convergence zone ($\approx$ 120 kyd)

Time of CPA: t = 0

OCEAN

Sound Speed Profiles:

Winter Depth (ft)	Winter Sound Speed (ft/sec)	Summer Depth (ft)	Summer Sound Speed (ft/sec)
0	4990.4	0	5052
420	4998.5	3440	4875
3720	4886.5	15660	5053
15540	5050		

Bottom Type: LRAPP bottom type 3

Table 3-1 continued.

ATMOSPHERE

Wind:

Wind speed: 8.8 kts

Direction: 45° with respect to source (receiver) track

Air Temperature:

Winter: Less than 50°F

Summer: Greater than 50°F

3.1.1 Summer

The program output for the summer profile is shown in Figs. 3-1 and 3-2 (CPA = 4 kyds). Rough surface effects are, in general, insignificant for the parameters of this example and will not be discussed. (Section 3.2 presents an example for which wind effects are substantial and are elaborated upon in detail.)

All results are plotted against time on the abscissa. The time shown in the figure is the difference, in secs, between the arrival time of the signal and the arrival time of the direct ray from the source at the CPA point. This latter arrival time is merely an arbitrarily selected time origin. It is generally very close to the time of CPA, if the CPA distance is not exceedingly large. With the source/receiver track configuration assumed, negative time differences occur when the source is approaching the receiver and positive differences occur when the source is moving away from the receiver.

Also shown on the abscissa are two ranges, both in kyds = (1) $R'(t)$, the horizontal source/receiver range at the time the sound energy leaves the air source and (2) $R'(t_A)$, the horizontal source/receiver range at the time the signal arrives at the receiver. Since a signal leaving the source at a given time may travel to the receiver by different propagation modes, with different propagation times, $R'(t_A)$ may differ between the modes.

The upper portion of Fig. 3-1 displays the transmission loss time history, where the transmission loss is referred to the air source level at 1 yd from the source. To obtain absolute intensity levels, the air source level, at 1 yd, must

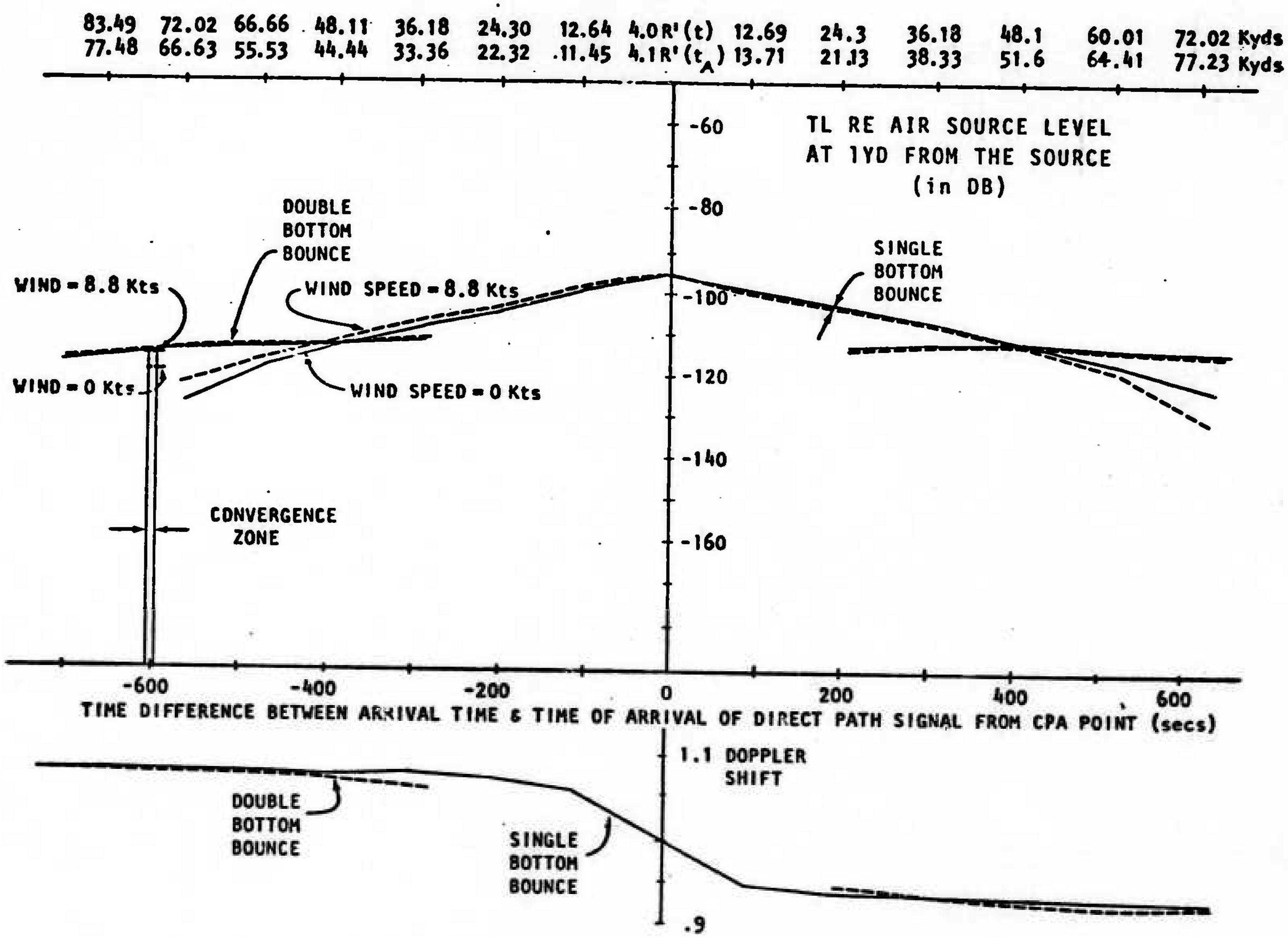


FIGURE 3-1 TIME HISTORIES OF TRANSMISSION LOSS AND NORMALIZED RECEIVED FREQUENCY, SUMMER PROFILE.

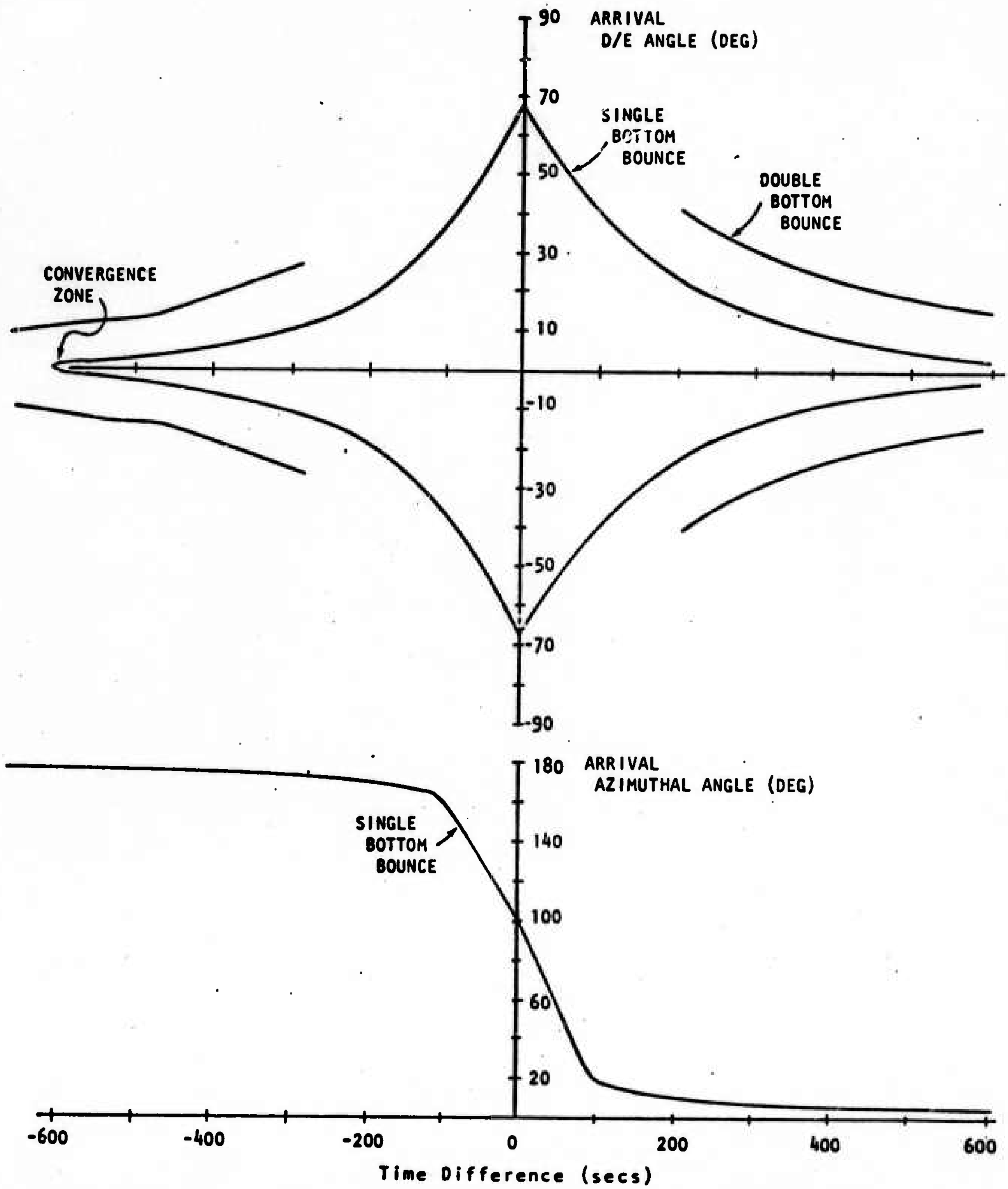


FIGURE 3-2 TIME HISTORIES OF ARRIVAL ANGLES, SUMMER PROFILE

be added to the TL values shown. The TL for each propagation mode is shown separately, both for the smooth surface conditions (wind speed = 0 kts) and for the most probable sea state (wind speed = 8.8 kts). Three propagation mode types are identifiable:

- single bottom bounce
- convergence zone
- double bottom bounce.

The direct path, if it occurs, will be important only near the time origin (near CPA time) and, although always calculated by the program, should be ignored for horizontal ranges beyond several kyds since for ranges greater than this, the direct path mode cannot exist physically. When the direct path does not exist, bottom bounce and/or surface duct modes (if a duct is present) must be considered. Note that although the direct path mode does not actually exist, we have still used the arrival time of the direct path from the CPA point as an arbitrary time origin. One could just as well plot all results versus the actual arrival time.

For the summer profile, a surface duct is not present, and bottom bounce modes will be the only available propagation paths from times near CPA time out to the times corresponding to the convergence zone arrival. In most cases, the single bottom bounce mode will dominate until the convergence zone arrival appears. However, for this example, the double bottom bounce arrival is significant even within the convergence zone because the bottom loss is almost negligible for the grazing angles associated with this arrival. While the bottom loss for the single bottom bounce arrival is also small for arrivals originating near the convergence zone, the spreading loss increases rapidly as the range approaches the convergence zone

range. Thus, in this example, the single bottom bounce arrival becomes weaker than the double bottom bounce arrival as the inner edge of the convergence zone is approached. The single bounce signal eventually disappears completely within the convergence zone.

In this example, the sound speed excess is very small (c_b almost equal to c_s). Consequently, the convergence zone width is very narrow and the associated transmission loss relatively large, as a result of the fact that only a narrow tube of rays emitted from the virtual surface source can reach the zone. If the sound speed excess were greater, the zone width would be broader and the TL would be less (in magnitude).

Beyond the convergence zone, the single bottom bounce mode does not exist and double bottom bounce arrivals will generally dominate. In the current example, the signal level of the double bounce mode is rather high because the bottom losses are negligible.

It may be observed that there is an asymmetry in the TL time histories about the time origin, which becomes more pronounced for larger times. This effect is a consequence of the ray travel time.

The time history of the Doppler shift (ratio of received frequency to source frequency) is shown below the TL time history. The shift will, in theory, be different for each propagation mode. For large times, however, the shifts for all modes will be essentially the same because of the large distances involved, as the example illustrates. As the CPA point is approached, the shifts usually differ. Generally, the bottom bounce shift will be more gradual than the direct path shift, if a direct path exists. This is a consequence

of the fact that the effective source for the bottom bounce arrivals lies at one ocean depth below the ocean bottom for the single bounce mode (three ocean depths below the bottom for the double bounce arrivals).

The D/E arrival angle time history (Fig. 3-2) displays the angles mode by mode. Positive angles correspond to surface-reflected arrivals, negative to direct arrivals.

Azimuthal arrival angle time histories (bottom of Fig. 3-2) generally display very gradual changes for large times and a rather rapid transition near the CPA time. Because of ray travel time differences, the time histories near the time origin may be mode-dependent.

3.1.2 Winter

Results for the winter profile are displayed in Figs. 3-3 and 3-4.

The winter profile has a surface duct and can therefore permit an additional propagation mode, the surface duct mode, which may be important if the receiver depth is less than the surface duct depth. In the present example, the receiver is in the surface duct, but the diffractive losses due to below-layer leakage are extremely large. Consequently, the surface duct TL is greater than 200 dB (in magnitude) and this mode may be ignored. In general, surface duct propagation will be negligible for low frequencies, as illustrated by the example.

The only essential difference between the winter and summer profile results arises from the fact that the sound speed excess is considerably larger for the winter profile. Consequently, the convergence zone is broader and the TL less (in magnitude).

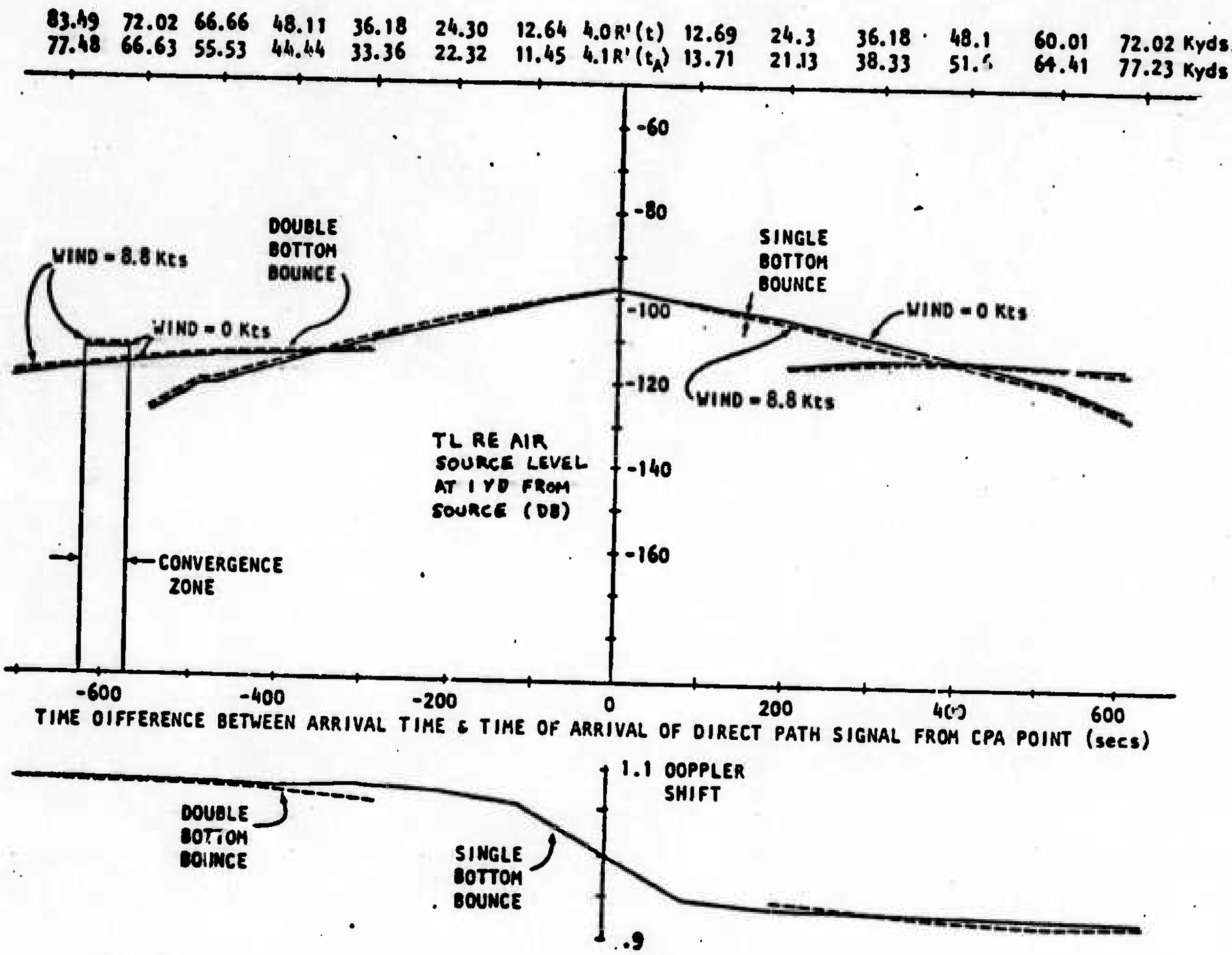


FIGURE 3-3 TIME HISTORIES OF TRANSMISSION LOSS AND NORMALIZED RECEIVED FREQUENCY, WINTER PROFILE.

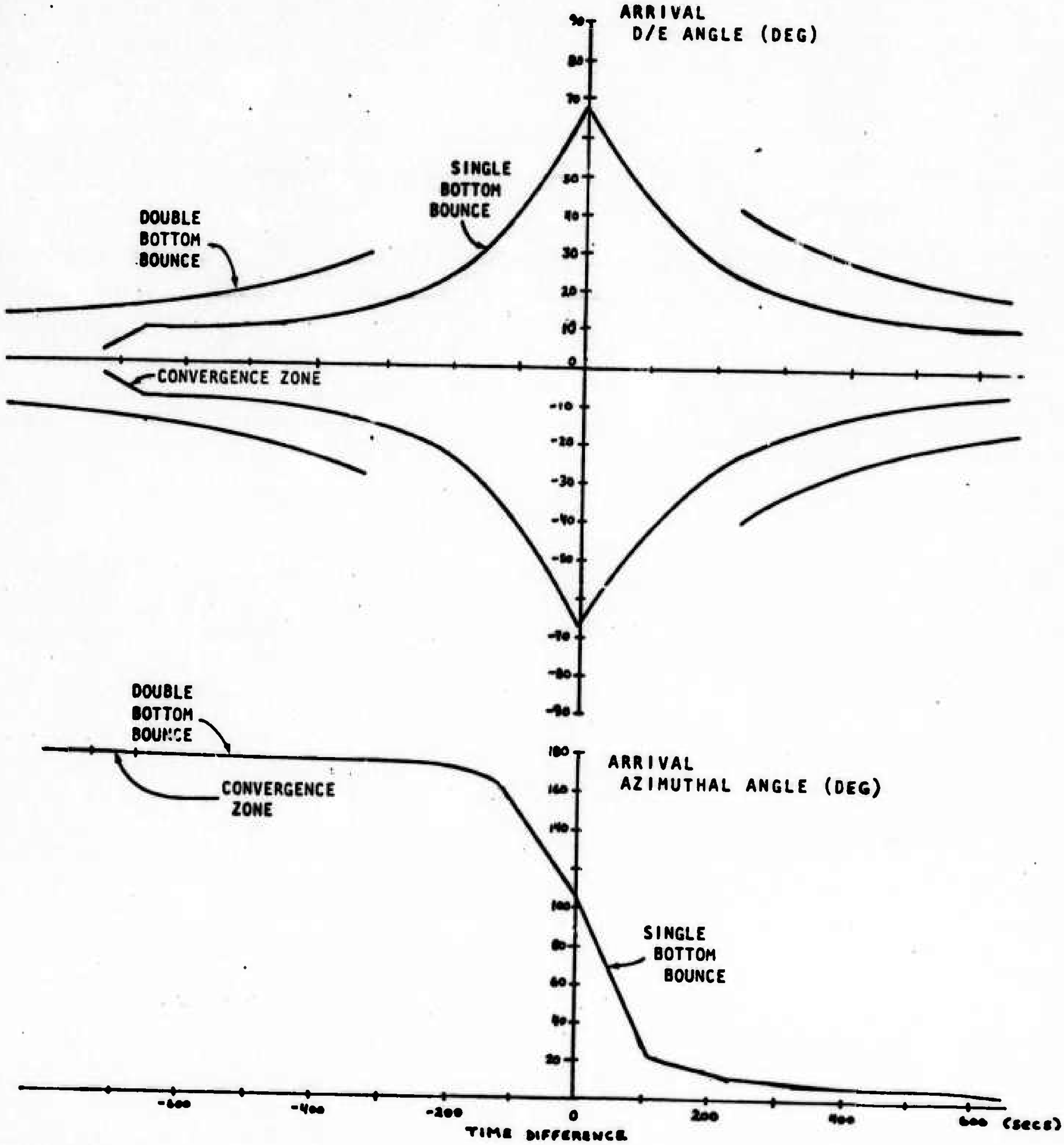


FIGURE 3-4 TIME HISTORY OF ARRIVAL ANGLES, WINTER PROFILE.

3.2 Rough Air-Water Interface

Figure 3-5 presents the results of the algorithm for a case in which rough surface effects are significant. (These results were computed by hand at any early stage of the program development. All the equations used are the same as in the programmed algorithm; however, the results may differ in some minor details from the program output because of calculational errors in the hand calculation. A limited set of spot checks indicates that, on the whole, the results shown in Fig. 3-5 and the program output will essentially agree.) In this example, the direct path does exist close to the receiver (within several kyds) since the CPA is 1/3 kyd.

Arrival angle and Doppler time histories are unchanged by rough surface effects. In contrast, the TL levels may be substantially different, although the general shape of the TL history is unchanged except in the direct path region (see, for example, the 1000 ft source height results). A number of effects are operating simultaneously to produce the differences in the TL curves. These are summarized in Table 3-2. (See Ref. 1 for a more detailed discussion of the table parameters.) In the following, each of the TL curves will be considered in turn, and the effects of the parameters in Table 3-2 will be discussed. The virtual source plots given in Ref. 1 should be kept in mind.

Consider first the 50 ft source height curve, with the wind equal 24.5 kts. In the double bottom bounce regions, the emission angles (at the virtual surface source) are moderately small (15° - 18°) and the effects of the most probable slope are still important. With the assumed wind and source/receiver track configuration, the receiver sees an arrival which has been emitted parallel to the wind for negative times (source

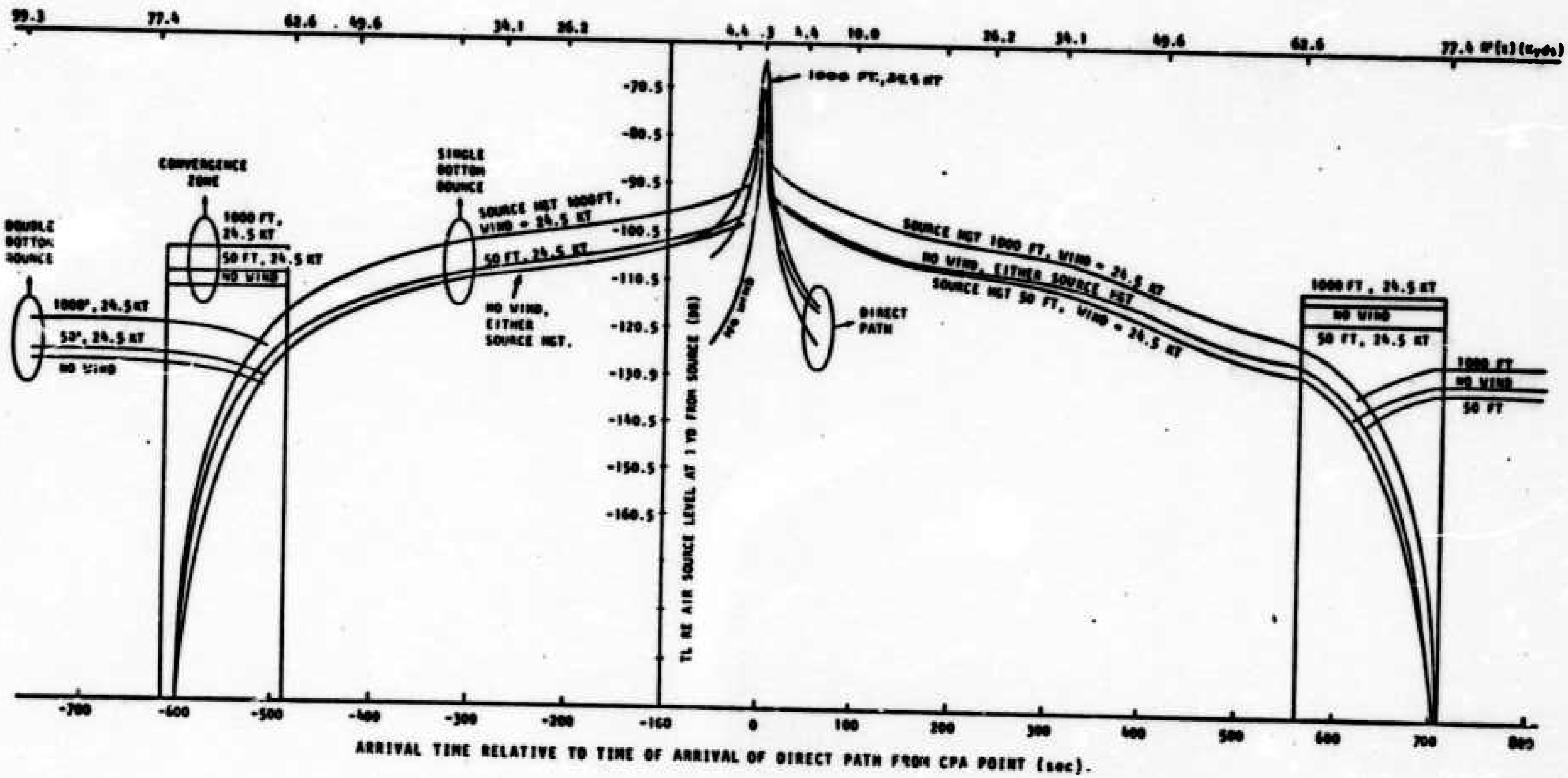


FIGURE 3-5(a) TRANSMISSION LOSS TIME HISTORY, HIGH SEA STATE
CPA = 1/3 Kyd, Air Vehicle Speed = 220 Kt, Receiver Speed = 10 Kts, Parallel Tracks. Air Vehicle Traveling Downwind. Source Frequency = 1000 Hz. Summer SVP, Receiver Depth = 200 Ft. LRAPP Bottom Type 3 Assumed.

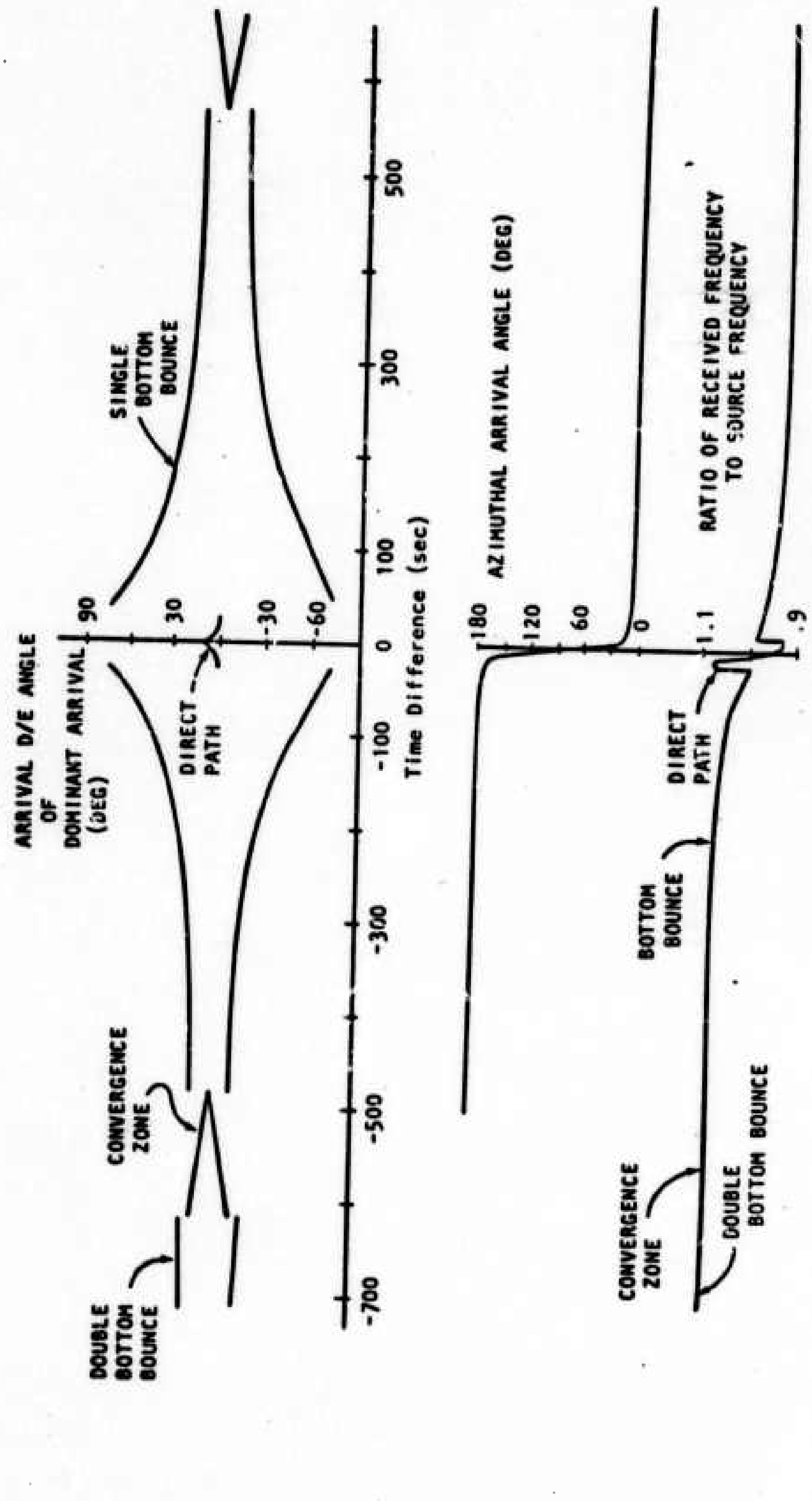


FIGURE 3-5(b) TIME HISTORIES OF ARRIVAL ANGLE AND RECEIVED FREQUENCY, HIGH SEA STATE

TABLE 3-2

FOR GIVEN SEA STATE, TRANSMISSION GAIN RELATIVE TO A SMOOTH INTERFACE DEPENDS ON:

- θ - Depression angle of ray leaving virtual source
- ϕ_W - Angle between plane of acoustic path and direction of waves
- f - Source frequency
- h - Source height
- σ - RMS slope (effective)
- L - Correlation length of surface slopes (effective)
- $\bar{N}$ - Average number of specular paths
- ψ - Most probable slope

approaching receiver) while the arrival for positive times (source moving away from the receiver) was emitted anti-parallel to the wind. The intensity of the outgoing ray at the virtual source is consequently somewhat greater than the flat surface intensity for negative times and somewhat smaller for positive times. The average number of specular paths is about unity for this low source height. Thus the TL lies slightly above the flat surface TL for negative times and slightly below for positive times.

In the single bottom bounce domain, the emission angles are moderately small for large negative or positive times and increase as the source/receiver range decreases (smaller negative/positive times). For large negative/positive times, the same effects operate as in the double bottom bounce case. As the time origin is approached, the emission angles get very steep and the rough surface effects diminish. Thus the TL approaches flat surface values as the source/receiver range diminishes.

Going back to the convergence zone regions, it can be seen from the D/E angle time histories that emission angles from 0° to 13° are involved. All the energy emitted by the source in this angular range appears in the zone. For large negative times, the received arrivals are emitted parallel to the wind and hence a modest gain, over the smooth surface, is seen; for large positive times, the arrivals are emitted anti-parallel to the wind and a modest loss, relative to smooth surface, is the result. As in the bottom bounce case, the average number of specular paths is about unity.

The direct path region displays the effects of the most probable slope most strongly. Over most of the direct path region, only very small emission angles are involved. Furthermore, the angle between the plane of the acoustic path and the wind changes

rapidly. At the outer edge of the region for negative times, the receiver sees rays emitted at very small depression angles in a direction almost parallel to the wind. Consequently, there is a large gain relative to the smooth interface results. At the other edge for positive times, this gain is substantially diminished because the arrivals are emitted in the direction anti-parallel to the wind. As the CPA point is approached, the received rays are emitted more and more in a direction perpendicular to the wind and the effect of the most probable slope becomes increasingly less pronounced. In addition, the emission angle increases. Hence only slight gains are predicted.

When the source height is increased, the effective rms slope decreases somewhat, and in the absence of any other effects, the gains would actually decrease relative to the 50 ft case. However, the average number of specular paths goes up substantially and results in generally large gains, relative to the smooth surface, for all times* This effect is most pronounced for negative times, where the received rays are emitted parallel to the wind, and the favorable disposition of the most probable slope further enhances the gains due to the large $\overline{N}$. For positive times, where the received rays are emitted anti-parallel to the wind, the effect of the multiple specular paths is diminished by the unfavorable orientation of the most probable slope in regions where the emission angles are small (direct path and convergence zone), and only slightly affected in the bottom bounce regions, where the emission angles are larger.

* The formalism used to calculate the average number of specular paths is probably not correct when the Fresnel zone size is on the order of, or greater than the surface slope correlation length. If this is the case for the parameters of Fig. 3-5(a), then the decrease in transmission loss for the 1000 ft source height may not be correct. The resolution of this problem requires further investigation of the model. A study is currently in progress.

For very large source heights, the effective rms slope would become very small, and the average number of specular paths again approaches unity since the Fresnel zone size increases faster than the correlation length of surface slopes. The surface, in effect, again looks smooth and the TL should approach the smooth surface results.* Physically, the most probable slope should also vanish for large heights (large Fresnel zone size); however, in the algorithm, the most probable slope depends only on wind speed and hence some residual effects for small emission angles will still be predicted. The correrion of this artifact must await a suitable theory for the most probable slope.

* Provided that the acoustic Rayleigh parameter for the sea surface is small. Further development of the model to account for scattering (as opposed to specular transmission) is currently underway.

REFERENCES

1. D.A. Sachs, "AUSEX Air-Water Acoustic Propagation Model," Bolt Beranek and Newman Inc., TM W311(February 1976).

2. G. Hicks, F. Jackson, J. Nitsche, "A Summary of U.S. and U.S.S.R. Aircraft Noise Signature Data for the ARPA Aircraft Undersea Experiments (U)," TM W290 (October 1975 (Secret)).

3. J.E. Barger and D.A. Sachs, "Transmission of Sound Through the Scaled Ocean Surface," Bolt Beranek and Newman Inc., Report 3103a (August 1975).

4. I. Dyer and D.A. Ranken, "Preliminary Interim Summary of FNWC Propagation Loss Tables (DF 68) (U)," General Oceanology, Inc., TM GO-105 (20 July 1970) (Confidential).

5. "Acoustic Environmental Scenarios and Predictions for ASW (C)," Maury Center for Ocean Science, MC Report 011, Vol. 1 (October 1972) (Confidential).

APPENDIX A

SURFACE DUCT AND CONVERGENCE ZONE SUBROUTINE

Introduction and Summary

The program described below was written to calculate the transmission loss (and some other pertinent variables) between a moving source above the surface of the water and a moving receiver lying beneath the surface. The program calculates (1) the transmission loss for sound rays which travel in a surface duct, if one is present, when the receiver is in the duct, and (2) the transmission loss for sound rays which travel to the first convergence zone when the receiver is within the convergence zone. For both cases, the program also computes arrival angles and the Doppler shift of the arrival.

The program user supplies the parameters which describe the positions and velocities of the source and receiver at the time origin ("initial") as well as the conditions of the air, water and surface between them. The program then determines the propagation modes which apply and calculates the appropriate transmission loss and other transmission variables. A number of outputs are printed for the user. These include, besides the TL, various arrival times, the depression/elevation angle, azimuthal angle, Doppler shift, and horizontal and slant ranges for the pertinent propagation mode. If the surface is rough, both smooth and rough surface results are given. The program is driven by supplying time as the continuing parameter. It is possible to have certain spans of time during which some propagation modes are not possible. In this case the transmission loss must be considered to be infinite and no results are printed. This situation arises when (1) there is no surface duct, or (2) there is a surface duct but the range is too close

(<5280 feet) for the surface duct algorithm used to be valid, and (3) the source and receiver are too close or too far for convergence zone propagation.

A.1 Glossary

Input

Wind:
- speed in knots (WS)
- x,y direction cosines; $-1 \leq 0 \leq 1$ (XWC,YWC)

Air source:
- x,y components of velocity in kts (VAX,VAY)
- initial x,y position (at t=0) in ft (XIA,YIA)
- source height in feet (H)
- frequency in Hz (F)

Receiver:
- velocity along x-axis in kts (VS)
- initial x position (at t=0) in ft (XIS)
- depth in feet (D)
  (y coordinate of receiver always assumed to be zero)

Sound velocity profile:
- depths in ft (0, DS, DMIN, DB)
- velocities in ft/sec (CS, CMAX, CMIN, CB)

Sound velocities (ft/sec):
- CS speed at surface
- CMAX speed at SVP maximum
- CMIN speed at SVP minimum
- CB speed at ocean bottom

Depths (ft):
- DS depth of surface duct
- DMIN depth of SVP minimum
- DB depth to ocean bottom

Time:
- starting, ending and step size in seconds (TI, TF, DT)

Questions:
- is all input data already in computer? (DATAIN) Y(es) or N(o)

- is air temperature greater than 50°F? (QTEMP)
  Y(es) or N(o)
- is there a surface duct? (LANS)
  Y(es) or N(o)
- rough or smooth surface? (RSS)
  R or S

<u>Output</u>:

Propagation mode:
- SD surface duct, smooth surface
- SDR surface duct, rough surface
- CZS convergence zone with surface duct, smooth surface
- CZSR convergence zone with surface duct, rough surface
- CZ convergence zone, no surface duct, smooth surface
- CZR convergence zone, no surface duct, rough surface

Times (in secs):
- t, t_A, $t_A - t_{AMAX}$, t_0 (T, TA, TDIF, TO)

  t = time of emission

  t_A = arrival time of propagated energy

  $t_A - t_{AMAX}$ = difference in arrival times between arrival at t_A and arrival time of direct path from CPA

  t_0 = time of CPA

Ranges:
- R', R (in ft - RP, R; in kyds - RP3, R3)
- $R'_{t=t_A}$ (in kyd - RPA3)
- R_0 (in ft - RO)

Doppler shift:
- dimensionless (DOP)

Arrival angles:
- D/E (depression/elevation, in deg - DE)
- ϕ (azimuthal, in deg - PHI)

Transmission loss:
- squared pressure ratio re air source level (p_a^2) at 1 ft, smooth or rough surfaces (w/o atmospheric attenuation) (PRSQ, PRS)

- in dB, smooth or rough surfaces (w/o atmospheric attenuation) (TL, TLRS)
- in dB re air source level at 1 yd (including atmospheric attenuation) (TL3)

A.2 Description

The program consists of a main section plus numerous subroutines. The main section calculates some parameters and directs the flow of the calculations. The subroutines are used to input data, evaluate certain functions, and to calculate the transmission losses. Table A-1 gives a general outline of these subroutines. Details of the calculations performed in the main program and the subroutines which calculate transmission loss follow.

The following sections give further details on the calculations performed by the main program and its subroutines.

A.2.1 Main program

The first event in the main program is inputting the data. This may be done within the program or partially done by using a BLOCK DATA subroutine. In either case the following values will be input: C_1, C_2, n, wind speed, and $n_{[illegible]}$ for the wind, v_{ax}, v_{ay}, x_{1a}, y_{1a}, h, v_s, x_{1s}, D, C_S, C_b, C_o, C_{min}, and d_{min}.

The program also demands some non-numerical input. It must be told whether or not the air temperature is greater than 50°F, if a surface duct exists and if the surface of the sea is rough or smooth. If a surface duct exists, values must be supplied for C_{max} and d_S. Finally, values for the range of time to be covered, t_i, t_f, Δt, are input along with the frequency f. The following values are now calculated by the program:

TABLE A-1

NAME	DESCRIPTION
Subroutines which calculate transmission loss	
SD	Calculates t_a, t_a-t_{AMAX}, ϕ, D/E, doppler, slant range and horizontal range, pressure^2 ratio; and transmission loss for sound traveling in the surface duct when the sea surface is smooth.
SDR	Calculates the corrected values of pressure^2 ratio and transmission loss for sound traveling in the surface duct when the sea surface is rough.
CZS	Calculates t_a, t_a-t_{amax}, ϕ, doppler, slant range and horizontal range and location of the convergence zone in the presence of a surface duct for a smooth surface. If the receiver lies within the convergence zone, the subroutine also calculates D/E, the pressure^2 ratios and the transmission loss.
CZSR	Calculates the corrected values of pressure^2 ratio and transmission ratio when the surface is rough and there is a surface duct and the receiver lies within the convergence zone.
CZ	Calculates t_a, t_a-t_{amax}, ϕ, doppler, slant range and horizontal range, and the location of the convergence zone when the sea surface is smooth and there is no surface duct. If the receiver lies within the convergence zone, the subroutine also calculates D/E, the pressure^2 ratio and the transmission loss.
CZR	Calculates the corrected values of pressure^2 ratio and transmission loss when the surface is rough, there is no surface duct and the receiver lies in the convergence zone.
WOUT	Calculates corrected values of transmission loss by including the effects of the air path. This subroutine also converts ranges to kyds and gives the transmission loss re 1 yd.

Table A-1 continued.

NAME	DESCRIPTION

Subroutines which evaluate functions

INT — Evaluates the function $I(\theta_i, \bar{\psi}, \sigma_W, \phi_W)$,

$$I = \frac{s^3}{4}\Big[\frac{\theta_i}{s} + \frac{2}{3}\Big\{\Big(\frac{A_1}{s}\Big)^3 - \Big(\frac{A_2}{s}\Big)^3\Big\} + \mathrm{erf}(A_1)\Big\{1 + \frac{2}{3}\Big(\frac{A_1}{s}\Big)^2\Big\}\Big(\frac{A_1}{s} - \mathrm{erf}(A_2)\Big\{1 +$$

$$\frac{2}{3}\Big(\frac{A_2}{s}\Big)^2\Big\}\Big(\frac{A_2}{s}\Big) + \frac{1}{3\sqrt{\pi}}\, e^{-(A_1/s)^2}\Big\{5 + 2\Big(\frac{A_1}{s}\Big)^2\Big\} -$$

$$\frac{1}{3\sqrt{\pi}}\, e^{-(A_2/s)^2}\Big\{5 + 2\Big(\frac{A_2}{s}\Big)^2\Big\}\Big]$$

where $s = \sqrt{2}\,\sigma_w$, $A_1 = \theta_i + \bar{\psi}\cos(\phi_w)$,
and $A_2 = \bar{\psi}\cos(\phi_w)$

ERF — Evaluates the error function,

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$$

Subroutines which input data

WIND — Inputs wind speed in knots and the dimensionless x and y direction cosines of the wind.

AIRV — Inputs the aircraft velocity as vectors in the x and y directions. The velocities are input in knots and converted to ft/sec.

AIRC — Inputs the initial (i.e., t=0) x and y position of the air source.

AIRH — Inputs the height of the aircraft in ft.

AIRF — Inputs the frequency of aircraft radiation in Hz.

SUBV — Inputs the velocity of the receiver in the x direction in knots and converts to ft/sec. (The y-direction velocity is assumed to be zero.)

Table A-1 continued.

SUBC	Inputs the x position of the receiver at t=0 in ft. (The y-position is assumed to be zero.)
SUBD	Inputs the depth of the receiver in ft.
ENVS	Inputs the sound velocity at the surface of the sea in ft/sec.
ENVC	Inputs the sound velocity at the bottom in ft/sec. Also inputs the depth of the bottom in ft.
ENVD	Inputs the minimum sound velocity (deep sound channel) in ft/sec along with the depth at which it occurs in ft.
ENVA	Inputs the sound velocity at the bottom of the surface duct in ft/sec (maximum sound velocity). The depth of the surface duct is also input in ft.
TIM	Inputs the starting and ending times for the calculations as well as the step size in time. All these values are in seconds.
WATIN	Subroutine used to change values of the input and then rerun the program.

$$t_0 = \frac{(x_{ia} - x_{is})(v_s - v_{ax}) - y_{ia} v_{ay}}{(v_s - v_{ax})^2 + v_{ay}^2} \quad (1)$$

$$R'_{(t=t_0)} = \sqrt{\left[(v_s t_0 + x_{is}) - (v_{ax} t_0 + x_{ia})\right]^2 + \left[v_{ay} t_0 + y_{ia}\right]^2} \quad (2)$$

$$R_{(t=t_0)} = \sqrt{\left[R'_{(t=t_0)}\right]^2 + D^2} \quad (3)$$

If the surface is rough, the following values also are calculated:

$$\xi = \left(\frac{8h\,(c/f)\, g_F^2}{\pi^2 U_F^4} \right)^{\frac{1}{4}} \quad (4)$$

where g_F is the acceleration due to gravity in ft/sec^2 and U_F is the wind speed in ft/sec.

$$\sigma_w = \frac{1}{\sqrt{2}} \left\{ 1.15 \times 10^4 \sqrt{\frac{\pi}{2}}\, \frac{U_c}{g_c^3} \left[1 - \mathrm{erf}(\xi) \right] \right\}^{\frac{1}{2}} \quad (5)$$

where U_c and g_c are U_F and g_F converted to cgs units. The program also calculates

$$\bar{\eta}^2 = 2.3 \times 10^4 \frac{\sqrt{2}}{g_c U_c^2} \left(\frac{4}{3} \right) \left[\sqrt{\pi} \left\{ 1 - \mathrm{erf}(\xi) \right\} + \frac{e^{-\xi^2}}{\xi} \left\{ \frac{1}{2\xi^2} - 1 \right\} \right] \quad (6)$$

$$m = h_c \sqrt{\overline{\eta^2}} \tag{7}$$

where h_c is the source height in cgs units.

$$\text{coefficient of } \bar{N} = \frac{m^2}{4\pi}\left[e^{-1/m^2} + \frac{\sqrt{\pi}}{2m}\left\{1 + \text{erf}\left(\frac{1}{m}\right)\right\}\right]^2 2 \tag{8}$$

$$\overline{\Psi} = 2.86 \times 10^{-3} \times U_c \times \frac{\pi}{180} \tag{9}$$

where the error functions, erf, are evaluated by subroutine ERF.

All these parameters are not functions of time. The program now is ready to march through time. The parameter t is set equal to the starting value of time t_1 (no necessarily equal to zero; may be plus or minus) and the foll 6 parameters are calculated:

$$x_s = v_s t + x_{is} \tag{10}$$

$$x_a = v_{ax} t + x_{ia} \tag{11}$$

$$y_a = v_{ay} t + y_{ia} \tag{12}$$

$$R' = \sqrt{(x_s - x_a)^2 + y_a^2} \tag{13}$$

$$R = \sqrt{(R')^2 + D^2} \tag{14}$$

$$A = \left[\frac{0.1 f_K^2}{1 + f_K^2} + \frac{40 f_K^2}{4100 + f_K^2}\right]\frac{1}{3000} \tag{15}$$

where f_K is the frequency in kHz.

$$\theta_R = \cos^{-1}\left(\frac{c_s}{c_b}\right) \tag{16}$$

$$g_b = \frac{c_b - c_{min}}{d_b - d_{min}} \tag{17}$$

If there is a surface duct, the program will check to see if the receiver is within it. If the receiver is not in the surface duct the program skips to the convergence zone calculation. If the receiver does lie in the surface duct, the SD subroutine will be called, and the calculated values will be typed out. The subroutine WOUT will be called next to convert the values just calculated to other units and to include the effects of air losses. If the sea surface is rough, the subroutine SDR will now be called to calculate the effects of this roughness on the transmission loss. WOUT will be called again to operate on this new transmission loss.

The program next will go to the convergence zone calculation. If there is a surface duct, subroutine CZS will be called. In the absence of a surface duct, CZ will be called. In either case the subroutine returns a tag which indicates whether or not the receiver lies within the convergence zone.

If the tag says no, the program skips further calculations and chooses a new value of t. If the receiver is in the convergence zone the calculated parameter are typed out and WOUT is then called. If the sea surface rough the subroutine CZSR or CZR will be called, depending on the presence or absence of a surface duct. The values calculated will be typed out and WOUT will again be called.

Next a new value of t is calculated by adding Δt to the present value. This value is checked against the final value, t_f, and if it is smaller or equal to it, the program will start calculating again at the point where dependence on t begins. If the new value of t is beyond the final value allowed for the time parameter, subroutine WATIN is called. This subroutine allows changes to be made in many of the input parameters at which point the program may be continued. WATIN also allows the program to end, if that is the desired option.

A.2.2 SD subroutine

The SD subroutine calculates the following:

$$t_a = t + \left(\frac{h}{c_1}\right) + \left(\frac{R'}{c_2}\right) \tag{18}$$

$$t_a - t_{t_{max}} = t - t_0 + \frac{(R' - R_{(t_0 t_0)})}{c_2} \tag{19}$$

$$\phi = \cos^{-1}\left(\frac{x_a - x_a}{R'}\right) \tag{20}$$

$$\text{doppler} = \sqrt{\frac{1 - \frac{2v_a}{c_2}\left(\frac{x_a - x_a}{R}\right)}{1 - \frac{2v_a}{c_2}\cos(\psi)}} \tag{21}$$

where $$\cos(\psi) = \frac{v_{ax}(x_s - x_a) - v_{ay} y_a}{v_a R} \tag{22}$$

$$D/E = \sqrt{\frac{2(C_{max} - C_s)}{C_{max}}} \tag{23}$$

$$A_{tot} = A + \left[\frac{7.44 \times 10^5}{f_k^{2/3} g_1^{4/3} d_s^2} + SSM\sqrt{\frac{f_k}{d_s}}\right] \tag{24}$$

where $$SSM = \begin{Bmatrix} 9.0 \\ 6.75 \\ 4.5 \end{Bmatrix} \text{ depending on } U_k \text{ (wind speed in kts)} = \begin{Bmatrix} >10 \\ 7\text{-}10 \\ <7 \end{Bmatrix} \text{ KTS.} \tag{25}$$

and $$g_1 = \frac{C_{max} - C_{min}}{d_{min} - d_s} \tag{26}$$

$$\frac{p_d^2}{p_a^2} = \frac{8n^2}{3R'd_s}\left(\frac{2[C_{max} - C_s]}{C_{max}}\right)^{3/2} 10^{-\left(\frac{A_{tot} R'}{10}\right)} \tag{27}$$

$$TL = 10 \log_{10}\left(\frac{p_d^2}{p^2}\right) \tag{28}$$

A.2.3 SDR subroutine

This subroutine uses the values calculated by SD as well as the original input to calculate the following:

$$\cos(\phi_w) = \frac{n_x(x_s - x_a) - n_y y_a}{R'} \tag{29}$$

$$\bar{N} = (\text{coefficient of } \bar{N})\left\{ \operatorname{erf}\left(\frac{\tan\left(\frac{D/E}{2}\right)}{\sqrt{2}\ \sigma_w}\right) + \operatorname{erf}\left(\frac{\left[\frac{\left(1-\sin\left(\frac{D/E}{2}\right)\right)}{\cos\left(\frac{D/E}{2}\right)}\right]}{\sqrt{2}\ \sigma_w}\right)\right\} \tag{30}$$

$$\frac{p_{SDR}^2}{p_i^2} = \frac{8 n^2}{R' d_s}\ I\left(D/E, \bar{\Psi}, \sigma_w, \phi_w\right) \bar{N} \times 10^{-\left(\frac{A_{bot} R'}{10}\right)} \tag{31}$$

$$TL = 10 \log_{10}\left(\frac{p_{SDR}^2}{p_i^2}\right) \tag{32}$$

where $I(D/E, \bar{\Psi}, \sigma_w, \phi_w)$ is evaluated by using subroutine INT and the error functions are evaluated in ERF.

A.2.4 CZS subroutine

The first part of this subroutine performs the same calculations listed under the SD subroutine, equations (18) through (23) and (26). In addition, the following are calculated:

$$g_s = \frac{c_{max} - c_s}{d_s} \tag{33}$$

$$R_{sb} = 2 c_{max}\left[\frac{1}{g_s}\sqrt{1-\left(\frac{c_s}{c_{max}}\right)^2} + \left(\frac{1}{g_1} + \frac{1}{g_b}\right)\sqrt{1-\left(\frac{c_{min}}{c_{max}}\right)^2}\right] \tag{34}$$

$$R_b = 2 c_b\left[\frac{1}{g_s}\sqrt{1-\left(\frac{c_s}{c_b}\right)^2} + \left(\frac{1}{g_1} + \frac{1}{g_b}\right)\sqrt{1-\left(\frac{c_{min}}{c_b}\right)^2} - \left(\frac{1}{g_s} + \frac{1}{g_1}\right)\sqrt{1-\left(\frac{c_{max}}{c_b}\right)^2}\right] \tag{35}$$

Now the program can decide if the receiver is in the convergence zone or not. If the following criterion is not met, the tag 'EXIST' will be no, 'N', and the subroutine will return to the main program. If

$$R_{SD} \leq R' \leq R_B \quad \text{or} \quad R_B \leq R' \leq R_{SD} \tag{36}$$

then 'EXIST' will be 'Y' (yes) and the following calculations are performed:

$$\theta_{SD} = \cos^{-1}\left(\frac{c_s}{c_{max}}\right) \tag{37}$$

$$t_1 = t_0 + \frac{t}{|t|}\sqrt{t_0^2 + \left[\frac{(x_{ia} - x_{ia})^2 + y_{ia}^2 + R_B^2}{(v_A - v_{ax})^2 + v_{ay}^2}\right]} \tag{38}$$

$$t_2 = t_0 + \frac{t}{|t|}\sqrt{t_0^2 + \left[\frac{(x_{ia} - x_{ia})^2 + y_{ia}^2 + R_{SD}^2}{(v_A - v_{ax})^2 + v_{ay}^2}\right]} \tag{39}$$

$$D/E = \theta_B + \frac{(t - t_1)}{(t_2 - t_1)}(\theta_{SD} - \theta_B) \tag{40}$$

$$\frac{p_{c2}^2}{p_1^2} = \frac{32n^2}{3}\,\frac{(\theta_B^3 - \theta_{SD}^3)}{(\theta_B + \theta_{SD})|R_B^2 - R_{SD}^2|}\,10^{-\left(\frac{\alpha R'}{10}\right)} \tag{41}$$

$$TL = 10\log_{10}\left(\frac{p_1^2}{p_2^2}\right) \tag{42}$$

A.2.5 CZSR subroutine

This subroutine uses values calculated in subroutine CZS as well as data from the main program. The following values are calculated:

$$\cos(\phi_w) = \frac{n_x(x_a - x_a) - n_y y_a}{R'} \tag{43}$$

$$\bar{N} = (\text{coefficient of } \bar{N})\left\{\operatorname{erf}\left(\frac{\tan\left(\frac{\theta_{SD}+\theta_B}{2}\right)}{\sqrt{2}\,\sigma_w}\right) + \operatorname{erf}\left(\frac{\left[\frac{1-\sin\left(\frac{\theta_{SD}+\theta_B}{2}\right)}{\cos\left(\frac{\theta_{SD}+\theta_B}{2}\right)}\right]}{\sqrt{2}\,\sigma_w}\right)\right\} \tag{44}$$

$$\frac{p_{cz}^2}{p_a^2} = \frac{32n^2\left\{I(\theta_B,\bar{\psi},\sigma_w,\phi_w) - I(\theta_{SD},\bar{\psi},\sigma_w,\phi_w)\right\}}{\left[\theta_{SD}+\theta_B\right]\left|R_B^2 - R_{SD}^2\right|}\,\bar{N}\times 10^{-\left(\frac{AS'}{10}\right)} \tag{45}$$

$$TL = 10\log_{10}\left(\frac{p_{cz}^2}{p_a^2}\right) \tag{46}$$

A.2.6 CZ subroutine

The first part of this subroutine performs the same calculations listed under the SD subroutine, equations (18) through (23). Then the following are calculated:

$$g_s = \frac{c_s - c_{min}}{d_{min}} \tag{47}$$

$$R_B = 2c_b\left[\left(\frac{1}{g_a}+\frac{1}{g_b}\right)\sqrt{1-\left(\frac{c_{min}}{c_b}\right)^2}-\frac{1}{g_a}\sqrt{1-\left(\frac{c_s}{c_b}\right)^2}\right] \tag{48}$$

$$R_{(\theta_s=0)} = 2c_s\left(\frac{1}{g_a}+\frac{1}{g_b}\right)\sqrt{1-\left(\frac{c_{min}}{c_s}\right)^2} \tag{49}$$

Now the subroutine can decide whether or not the receiver is in the convergence zone. If it isn't, the tag, 'EXIST' is set to 'N', no, and control returns to the main program. If one of the following criteria are met:

$$R_B \leq R' \leq R_{(\theta_s=0)} \quad \text{or} \quad R_{(\theta_s=0)} \leq R' \leq R_B \tag{50}$$

then 'EXIST' is set to 'Y', yes, and the following are calculated:

$$t_1 = t_0 + \frac{t}{|t|}\sqrt{t_0^2-\left[\frac{(x_{ia}-x_{ia})^2+y_{ia}^2-R_B^2}{(v_a-v_{ax})^2+v_{ay}^2}\right]} \tag{51}$$

$$t_2 = t_0 + \frac{t}{|t|}\sqrt{t_0^2-\left[\frac{(x_{ia}-x_{ia})^2+y_{ia}^2-R_{(\theta_s=0)}^2}{(v_a-v_{ax})^2+v_{ay}^2}\right]} \tag{52}$$

$$D/E = \theta_B\left(1-\frac{(t-t_1)}{(t_2-t_1)}\right) \tag{53}$$

$$\frac{p_{cz}^2}{p_a^2} = \frac{32n^2\theta_B^2}{3\left|R_B^2-R_{(\theta_s=0)}^2\right|}10^{-\left(\frac{AR'}{10}\right)} \tag{54}$$

$$TL = 10\log_{10}\left(\frac{p_{cz}^2}{p_a^2}\right) \tag{55}$$

A.2.7 CZR subroutine

The values calculated in CZ are used in this subroutine along with values from the main program. The following are calculated:

$$\cos(\phi_W) = \frac{n_x(x_s - x_a) - n_y y_a}{R'} \tag{56}$$

$$\bar{N} = (\text{coefficient of } \bar{N}) \left\{ \operatorname{erf}\left(\frac{\tan\left(\frac{\theta_B}{2}\right)}{\sqrt{2}\,\sigma_W} \right) + \operatorname{erf}\left(\frac{\left[\frac{1 - \sin\left(\frac{\theta_B}{2}\right)}{\cos\left(\frac{\theta_B}{2}\right)} \right]}{\sqrt{2}\,\sigma_W} \right) \right\} \tag{57}$$

$$\frac{p_{czr}^2}{p_a^2} = \frac{32 n^2\, I(\theta_B, \bar{\Psi}, \sigma_W, \phi_W)}{\theta_B \left| R_B^2 - R^2_{(\theta_B = 0)} \right|} \bar{N} \times 10^{-\left(\frac{AR'}{10}\right)} \tag{58}$$

$$TL = 10 \log_{10}\left(\frac{p_{czr}^2}{p_a^2} \right) \tag{59}$$

A.2.8 WOUT subroutine

The following values are converted from ft to kyds:

$$R_3' = \frac{R'}{3000} \tag{60}$$

$$R_3 = \frac{R}{3000} \tag{61}$$

The value of R' is calculated at t_a and then expressed in kyds:

$$R'_{3(t=t_a)} = \left\{ \sqrt{(v_a t_a + x_{ia}) - (v_{ax} t_a + x_{ia})]^2 + [v_{ay} t_a + y_{ia}]^2} \right\} \frac{1}{3000} \tag{62}$$

The incoming TL is now corrected from dB re 1 ft to dB re 1 yd by the factor

$$TL_q = 10 \log_{10} (9) \tag{63}$$

and by a factor depending on temperature:

$$\begin{aligned} corr &= 1.25 \times 10^{-6} \times f \times h \quad \text{if temperature} < 50^\circ F \\ corr &= \frac{7.48 \times 10^{-8} \times f^2 \times h}{164.05} \quad \text{if temperature} > 50^\circ F \end{aligned} \tag{64}$$

The resulting transmission loss is:

$$TL_3 = TL + TL_q - corr \tag{65}$$

A.3 Directions for Running the Program

The main program and all the subroutines are collected together under the name WATER. It may be run in either of two ways:

1. The data may be entered by the user when running the program.

2. The data may be set up in a BLOCK DATA file. When the user is ready to run the program, the BLOCK DATA file is loaded at the same time as WATER. This option is useful if many parameters remain constant for different values of frequency and time.

An illustration of the first method is shown in Example 1. The user tells the computer to EXecute WATER F4. All user typing in the examples has been underlined. The computer now will compile the program, if the compiled version was not stored. Then the compiled version is loaded into the working area in the computer and the program is started.

The first question is whether the data has been included; since it hasn't, the user would type N or NO. Next the user must indicate if the air temperature is greater than 50°F. If the answer is to be "yes", the user would type Y or YES. Now the program will ask for several parameters:

- wind speed (kts) and direction cosines (dimensionless)
- aircraft velocity vectors (kts)
- aircraft (x,y) position and height above water (ft) at t=0
- submarine velocity (kts)
- submarine position and depth (ft) at t=0

The program now asks if there is a surface duct. The user says no by typing N or NO. In answer to the next question, the

EXAMPLE 1

```
.EX WATER.F4
LOADING

LOADER 5K CORE
6+3K MAX 526 WORDS FREE
EXECUTION

IS DATA IN COMPUTER? N

TEMPERATURE GREATER THAN 50 DEG F? Y

ENTER THE APPROPRIATE PARAMETERS IN
THE DIMENSIONS INDICATED.

WIND SPEED= 8.8
X DIRECTION COSINE OF WIND= .707
Y DIRECTION COSINE OF WIND= .707

VELOCITY VECTOR X-DIRECTION(KTS)= 220.
VELOCITY VECTOR Y-DIRECTION(KTS)= 0.

INTIAL X-COORDINATE OF AIRCRAFT( FT)= 0. *
INITIAL Y-COORDINATE OF AIRCRAFT(FT)= 12000.*

HEIGHT OF AIRCRAFT FROM SEA SURFACE(FT)= 10000.

VELOCITY VECTOR X-DIRECTION SUB(KTS)= 7.

INITIAL X-COORDINATE OF SUB(FT)= 0. *

DEPTH OF SUB(FT)= 400.

IS THERE A SURFACE DUCT? N
ROUGH OR SMOOTH SURFACE (R OR S)? S

SURFACE SOUND SPEED(FT/SEC)= 5052.

BOTTOM SOUND SPEED(FT/SEC)= 5053.
DEPTH FOR BOTTOM SOUND SPEED(FT)= 15660.

MIN. SOUND SPEED(FT/SEC)= 4875.
DEPTH AT MIN. SOUND SPEED(FT)= 3400.

INITIAL TIME= -1000.†
FINAL TIME= 1000.
TIME INCREMENTS= 25.

FREQ. OF AIRCRAFT RADIATION(HZ)= 150.
```

* Here, "initial" refers to t=0

† In this statement, "initial" refers to the starting time of the encounter.

```
SMOOTH SURFACE; NO SURFACE DUCT

T0=    0.0, R'(T0)= 12000., R(T0)= 12007., F=  150.000
H=10000.FT, WS= 8.8KTS, NX= 0.71, NY= 0.71

MODE    T     TA   TDIF      R'        R  DOP     D/F  PHI    P RATIO     TL
TO CHANGE A RUN PARAMETER, ENTER THE APPROPRIATE NUMBER:
   1  AIRCRAFT VEL. VECTORS
   2  AIRCRAFT INITIAL POSITION
   3  AIRCRAFT HEIGHT
   4  AIRCRAFT RADIATED FREQ.
   5  SUB VEL. VECTOR
   6  SUB INITIAL POSITION
   7  SUB DEPTH
   8  SURFACE SOUND SPEED
   9  BOTTOM SOUND SPEED AND DEPTH
  10 MINIMUM SOUND SPEED AND DEPTH
  11 MAX. SOUND SPEED AND DEPTH
  12 TIME OF EVENTS
  13 RUN
  14 STOP
  15 WIND PARAMETERS
ENTER THE APPROPRIATE NUMBER = 12

INITIAL TIME=  -1000.*
FINAL TIME=  1000.
TIME INCREMENTS=  5.


CHANGE PARAMETER = 13



SMOOTH SURFACE; NO SURFACE DUCT

T0=    0.0, R'(T0)= 12000., R(T0)= 12007., F=  150.000
H=10000.FT, WS= 8.8KTS, NX= 0.71, NY= 0.71

MODE    T     TA   TDIF      R'        R  DOP     D/F  PHI    P RATIO     TL
 CZ   -645.  -589.  -601.232216.232216. 1.081    0.46177.  1.304E-13-128.8
 CZ   -645.                77.405 77.405      70.759                  -119.4
 CZ   -640.  -585.  -596.230421.230421. 1.081    1.09177.  1.304E-13-128.8
 CZ   -640.                76.807 76.807      70.264                  -119.4
 CZ    640.   695.   684.230421.230421. 0.935    1.09   3. 1.304E-13-128.8
 CZ    640.                76.807 76.807      83.411                  -119.4
 CZ    645.   701.   689.232216.232216. 0.935    0.46   3. 1.304E-13-128.8
 CZ    645.                77.405 77.405      84.053                  -119.4

CHANGE PARAMETER = 14


EXIT
```

* Here, "initial" refers to the encounter starting time.

user types S to indicate a smooth sea surface (i.e., windspeed = 0) or R to indicate a rough surface (windspeed ≠ 0). Now the program will ask for the appropriate sound speed profile quantities:

- sound speed (ft/sec)
- at depth (ft)

Finally the program asks for the starting and final values of the time parameter, the step size in time, and the aircraft frequency of radiation. This concludes the data input and the calculations now proceeds. In the example illustrated, no results are outputted because the convergence zone is very narrow and the time step selected was too coarse.

After going through all values of time input, the program comes back to the user to ask if any changes in parameter are desired. If the user wants to change the times used, 12 must be typed, and then new values for the times used are entered. If no other changes are desired, the program can be rerun with the new values by typing 13. Some typical output is shown. Assuming that the user gets satisfactory results and no more changes are desired, the program may be stopped by typing 14. (An explanation of the output format is given in Section A.4.)

There is much less input at the time of running the program when BLOCK DATA is used. The BLOCK DATA file must be created in the format of Fig. A-1. Data are entered in the file using the same conventions and units as when the program is run without the file, with the following exceptions and additions:

1. source and receiver velocities are entered in ft/sec.

2. C1 is the sound speed in air (ft/sec)

Figure A-1. BLOCK DATA FORMAT

```
C
      BLOCK DATA
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      COMMON/MM/M
C
      DATA VAX/371.36/
      DATA VAY/0.0/
      DATA XIA/0.0/
      DATA YIA/12000.0/
      DATA H/10000.0/
      DATA F/150.0/
      DATA VS/11.816/
      DATA XIS/0.0/
      DATA D/400.0/
      DATA AXN/0.22/
      DATA WS/8.8/
      DATA XWC/.707/
      DATA YWC/.707/
C
      DATA C1/1100.0/
      DATA C2/5000.0/
      DATA CS/4990.4/
      DATA CB/5050.0/
      DATA DB/15540.0/
      DATA DS/420.0/
      DATA CMIN/4886.5/
      DATA CMAX/4998.5/
      DATA DMIN/3720.0/
      DATA ETA/1.0E-27/
      DATA PI/3.14159265/
C
      DATA M/1/
C
      END
```

3. C2 is the water sound speed in the vicinity of the surface (ft/sec). Any number close to 5000 ft/sec will be adequate.

4. ETA is a test number, against which the computed value of the ratio of the squared received pressure to squared source pressure at unit distance is compared before the TL is computed. If the ratio is less than ETA, the TL is set equal to -999.

5. M is an internally used tag which tells the computer to delete printing out the full list of data questions when M = 1.

In Example 2 the BLOCK DATA file is stored as B2, therefore execution of the program is begun by typing EX WATER.F4,B2.F4. In reply to the question on whether data is in the computer or not, the user would type Y or YES. The next question concerns the air temperature. If the temperature is less than 50°F the user would type N or NO.

The program now skips all the beginning input and asks if there is a surface duct. If there is, the user would type Y or YES. The user must be sure that this answer agrees with the sound speed profile included in the BLOCK DATA file. The type of sea surface is determined and then the program skips over the entry of sound speed profile data directly to the values of time and frequency to be used. In the example, an error was made in typing in the time increment. It was corrected by typing (control A) for each character to be deleted and then supplying the correct values. As soon as this data is entered, the calculations begin.

EXAMPLE 2

```
.EX WATER.F4,R2.F4
LOADING

LOADER 5K CORE
6+3K MAX 526 WORDS FREE
EXECUTION

IS DATA IN COMPUTER? Y

TEMPERATURE GREATER THAN 50 DEG F? N

IS THERE A SURFACE DUCT?  Y
ROUGH OR SMOOTH SURFACE (R OR S)? R

INITIAL TIME=  -660.
FINAL TIME=  -600.
TIME INCREMENTS=  50,05,30.

FREQ. OF AIRCRAFT RADIATION(HZ)=  150.


ROUGH SURFACE: SURFACE DUCT

T0=    0.0, R'(T0)= 12000., R(T0)= 12007., F=  150.000
H=10000.FT, WS= 8.8KTS, NX= 0.71, NY= 0.71

MODE     T    TA   TDIF       R'         R  DOP      D/F   PHI    P RATIO     TL
 SD   -660. -603. -615.237602.237603. 1.081    3.26177. 0.000E+00-999.0
 SD   -660.                79.201 79.201        72.425                  -999.0
 SDR  -660.                                         1.000E+00 0.000E+00-999.0
 SDR  -660.                                                             -999.0
 CZS  -660. -603. -615.237602.237603. 1.081    4.86177. 8.263E-13-120.8
 CZS  -660.                79.201 79.201        72.425                  -113.2
CZSR  -660.                                         1.000E+00 1.064E-12-119.7
CZSR  -660.                                                             -112.1
 SD   -630. -576. -587.226830.226831. 1.081    3.26177. 0.000E+00-999.0
 SD   -630.                75.610 75.610        69.094                  -999.0
 SDR  -630.                                         1.000E+00 0.000E+00-999.0
 SDR  -630.                                                             -999.0
 CZS  -630. -576. -587.226830.226831. 1.081    7.70177. 8.280E-13-120.8
 CZS  -630.                75.610 75.610        69.094                  -113.2
CZSR  -630.                                         1.000E+00 1.066E-12-119.7
CZSR  -630.                                                             -112.1
 SD   -600. -548. -559.216060.216060. 1.081    3.26177. 0.000E+00-999.0
 SD   -600.                72.020 72.020        65.762                  -999.0
 SDR  -600.                                         1.000E+00 0.000E+00-999.0
 SDR  -600.                                                             -999.0
```

CHANGE PARAMETER = 4

FREQ. OF AIRCRAFT RADIATION(HZ)= 5000.

CHANGE PARAMETER = 3

ROUGH SURFACE: SURFACE DUCT

```
T0=   0.0, R'(T0)= 12000., R(T0)= 12007., F= 5000.000
H=10000.FT, WS= 8.8KTS, NX= 0.71, NY= 0.71

MODE    T     TA    TDIF       R'         R   DOP      D/F  PHI     P RATIO      TL
 SD   -660. -603. -615.237602.237603. 1.081     3.26177. 6.400E-29-281.9
 SD   -660.               79.201 79.201        72.425                      -334.9
 SDR  -660.                                        1.085E+00 1.398E-28-278.5
 SDR  -660.                                                                -331.5
 CZS  -660. -603. -615.237602.237603. 1.081     4.86177. 1.798E-15-147.5
 CZS  -660.               79.201 79.201        72.425                      -200.4
CZSR  -660.                                        1.085E+00 2.514E-15-146.0
CZSR  -660.                                                                -199.0
 SD   -630. -576. -587.226830.226831. 1.081     3.26177. 3.406E-28-274.7
 SD   -630.               75.610 75.610        69.094                      -327.6
 SDR  -630.                                        1.085E+00 7.428E-28-271.3
 SDR  -630.                                                                -324.2
 CZS  -630. -576. -587.226830.226831. 1.081     7.70177. 2.379E-15-146.2
 CZS  -630.               75.610 75.610        69.094                      -199.2
CZSR  -630.                                        1.085E+00 3.324E-15-144.8
CZSR  -630.                                                                -197.7
 SD   -600. -548. -559.216060.216060. 1.081     3.26177. 1.817E-27-267.4
 SD   -600.               72.020 72.020        65.762                      -320.4
 SDR  -600.                                        1.085E+00 3.955E-27-264.0
 SDR  -600.                                                                -317.0
```

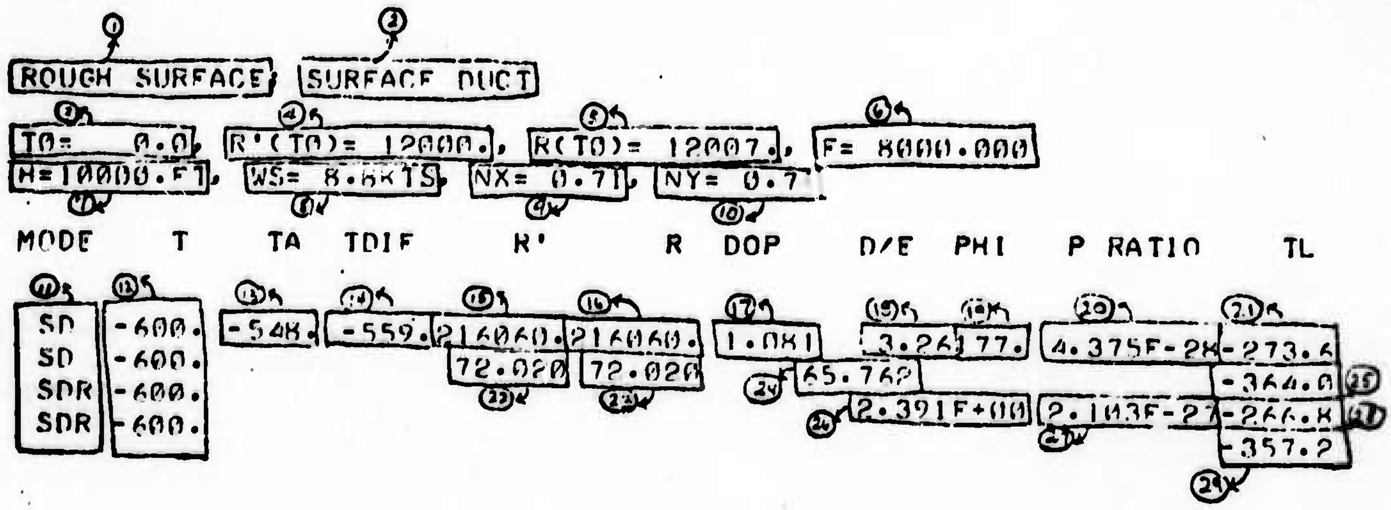
CHANGE PARAMETER = 14

In the example the transmission loss in the surface duct is given as -999. This is not the actual value, but is a convention used to show that the loss is so small it is not possible for the computer to work with such small numbers.

At completion of the time values given, the program asks if there are any parameter changes desired. If a different frequency is needed, the user would type 4 and then give the new value. If no more changes are desired, the user would tell the program to rerun by typing 13. After completion of this new output, the user would indicate the end of the run by typing 14 in answer to the change parameters question.

A.4 Output Description – 1

A description of the output produced by the program is given in detail below. An example is given on which each item has been tagged. These tags are used in the following explanation to indicate the item under discussion.



EXAMPLE

Output Description – 2

As soon as all input has been entered, the program prints out a heading which indicates whether the sea surface is smooth or rough (1), and if there is a surface duct or not (2). Next several parameters which describe the data as well as some parameters which do not depend on time are printed. t_0 is given in seconds (3), R' and R, calculated for $t = t_0$ are printed in ft, (4) and (5). Frequency is given in Hertz (6), aircraft height in ft (7). Finally the wind parameters are printed: wind speed in kts (8) and the x- and y-direction cosines, (9) and (10), which are dimensionless.

The program now types out a heading for the time dependent calculations. Next a line of data starts with a tag indicating the type of propagation (11). The names are SD for surface duct, CZS for convergence zone in the presence of a surface duct, and CZ for convergence zone when there is no surface duct. The addition of the suffix R means the rough sea surface case; the absence of a suffix refers to the smooth sea surface case. The next item in the line is the time used in seconds (12).

Data listed beyond these two items varies with the line. (In every case a minimum two lines of data are output. If there is a rough sea surface an additional two lines are output.)

In the first line the third item is t_A in seconds (13), followed by t_A-t_{AMAX} in seconds (14) R' in ft (15), and R in ft (16). Next come the doppler value (17) which is dimensionless, D/E in degrees (18), and azimuthal angle, ϕ, in degrees (19). Finally, the pressure ratio is printed (re 1 ft and without air losses) (20) and then the transmission loss (21) in dB re 1 ft (without air losses).

The third and fourth items in the second line are R' and R in kyds, (22) and (23). Next the range, calculated for $t = t_A$ is printed in kyds (24) and finally the transmission loss is printed (25). This transmission loss is in dB re 1 yd and is corrected to include air losses.

The third line only appears if the sea surface is rough. The third item in the line is $\bar{N}$ (26). Next the pressure ratio (re 1 ft and without air losses) is printed (27) and finally the transmission loss in dB re 1 ft without air losses (28).

The fourth line also appears only for rough sea surface conditions. The only term after the tag and time is the transmission loss in dB re 1 yd, corrected for air losses (29).

A.5 Program Listing

```
.TYPF WATFR.F4
C
C       TRANSMISSION LOSS CALCULATIONS  (100541)
C
        COMMON /IN/ VAX,VAY.XIA,YIA,H,F,VS,XIS.D,AXN,WS,XWC,YWC,TI.TF.DT
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSO,A,MODF
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CM  ,DMIN,ETA,GS,GB,RO,TO,PI,
        I G1
        COMMON/MM/M
        COMMON/RS/ SW,CPW,PSIB,FPN,CI,PRS,TLRS,BN
C
        C1=1100.0
        C2=5000.0
        AXN=0.22
        ETA=1.F-32
        PI=3.14159265
        SPI=SQRT(PI)
        AGSQ=32.174*32.174
        SWC=1.15E4*SPI/(SQRT(2.0)*(981.0**3))
        EBSC=2.3E4*4.0*SQRT(2.0)/(3.0*981.0)
C
        TYPF 987
987     FORMAT(' IS DATA IN COMPUTER? '$)
        ACCEPT 3,DATAIN
        TYPF 952
952     FORMAT(' TEMPERATURE GREATER THAN 50 DEG F? '$)
        ACCEPT 3,OTEMP
        IF ((DATAIN.EQ.'Y').OR.(DATAIN.EQ.'YES')) GO TO 985
        M=0
C
        TYPF 1
1       FORMAT(' ENTER THE APPROPRIATE PARAMETERS IN'/
        I' THE DIMENSIONS INDICATED.',/)
        CALL WIND(WS,XWC,YWC)
        CALL AIRV(VAX,VAY)
        CALL AIRC(XIA,YIA)
        CALL AIRH(H)
        CALL SURV(VS)
        CALL SURC(XIS)
        CALL SURD(D)
985     TYPF 2
2       FORMAT(' IS THERE A SURFACE DUCT?',2X,$)
```

```
      ACCEPT 3,LANS
3     FORMAT(A3)
      RSS='S'
      TYPE 981
981   FORMAT('+ROUGH OR SMOOTH SURFACE (R OR S)? '$)
      ACCEPT 3,RSS
      IF((LANS.EQ.'YES').OR.(LANS.EQ.'Y')) GO TO 4
      IF ((DATAIN.EQ.'Y').OR.(DATAIN.EQ.'YES')) GO TO 986
      CALL ENVS(CS)
      CALL ENVC(CB,DB)
      CALL ENVD(CMIN,DMIN)
      GO TO 5
4     IF ((DATAIN.EQ.'Y').OR.(DATAIN.EQ.'YES')) GO TO 986
      CALL ENVS(CS)
      CALL ENVC(CB,DB)
      CALL ENVD(CMIN,DMIN)
      CALL ENVA(CMAX,DS)
5     CONTINUE
986   CALL TIM(TI,TF,DT)
      CALL AIRF(F)
30    CONTINUE
C
      TO=((XIA-XIS)*(VS-VAX)-YIA*VAY)/(((VS-VAX)**2)+VAY*VAY)
      XO=VS*TO+XIS-VAX*TO-XIA
      YO=VAY*TO+YIA
      RPO=SQRT(XO*XO+YO*YO)
      RO=SQRT(RPO*RPO+D*D)
      IF (RSS.NE.'R') GO TO 740
      WSF=1.688*WS
      AL=CI/F
      XI=(8.0*H*AL*AGSQ/(PI*PI*(WSF**4)))**0.25
      CALL ERF(XI,FREXI)
      U=WSF*30.48
      SW=SQRT(SWC*U*(1.0-FREXI)/2.0)
      XI2=XI*XI
      FBS=FBSC*(SPI*(1.0-FREXI)+EXP(-XI2)*(1.0/(2.0*XI2)-1.0)
     1 /XI)/(U**3)
      SM=0.
      IF (FBS.GE.0.) SM=H*30.48*SQRT(FBS)
      OOM=1.E18
      IF (ABS(SM).GT.1.E-18) OOM=1.0/SM
      OOM2=OOM*OOM
      CALL ERF(OOM,FREIM)
      FOOM=0.0
      IF (OOM2.LT.60.) FOOM=EXP(-OOM2)
      FPN=0.5
      IF (SM.GT.1.E-18) FPN=(SM*SM*(FOOM+SPI*OOM*(1.0+FREIM)/
     1 2.0)**2)/(2.0*PI)
      PSIB=(2.86E-3)*U*PI/180.0
CCCC
C     TYPE 101,XI,SW,SM,PSIB
101   FORMAT(' XI=',1PE10.3,', SIG(W)=',1PE10.3,', M=',1PE10.3,
     1 ', PSIB=',1PE10.3)
C     TYPE 105,OOM,FBS,FOOM,FPN
105   FORMAT(' OOM=',1PE10.3,', FBS=',1PE10.3,', FOOM=',1PE10.3,
     1 ', FPN=',1PE10.3)
C
      IF ((LANS.EQ.'YES').OR.(LANS.EQ.'Y')) GO TO 745
      TYPE 980
980   FORMAT(//' ROUGH SURFACE: NO SURFACE DUCT'/)
      GO TO 751
745   TYPE 979
979   FORMAT(//' ROUGH SURFACE: SURFACE DUCT'/)
      GO TO 751
C
740   IF ((LANS.EQ.'YES').OR.(LANS.EQ.'Y')) GO TO 750
```

```
      TYPE 701
701   FORMAT(//' SMOOTH SURFACE; NO SURFACE DUCT'/)
      GO TO 751
750   TYPE 702
702   FORMAT(/' SMOOTH SURFACE; SURFACE DUCT'/)
751   TYPE 706,TO,RPO,RO,F
706   FORMAT(' TO='F6.1,', R''(TO)='F7.0,', R(TO)='F7.0,', F=',F9.3/)
      TYPE 763,H,WS,XWC,YWC
763   FORMAT('+H='F6.0,'FT, WS='F4.1,'KTS, NX='F5.2,', NY='F5.2/)
      TYPE 703
703   FORMAT(1H ,'MODE     T     TA  TDIF      R''     R  DOP     D/F'
     1  ' PHI   P RATIO    TL')
C
      T=TI
700   XS=VS*T+XIS
      XA=VAX*T+XIA
      YA=VAY*T+YIA
      XSA=XS-XA
      RP=SQRT(XSA*XSA+YA*YA)
      R=SQRT(RP*RP+D*D)
      FK2=F*F*(1.0E-6)
      A=((0.1*FK2)/(1.0+FK2)+(40.0*FK2)/(4100.0+FK2))/3000.0
      IF (RP.LT.5280.0) GO TO 900
      THEB=ACOS(CS/CB)
      GB=(CB-CMIN)/(DB-DMIN)
      IF (.NOT.((LANS.EQ.'Y').OR.(LANS.EQ.'YES'))) GO TO 800
      IF(D.GT.DS) GO TO 760
      CALL SD(XSA,YA,ARPM)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DF,PHI,PRSQ,TL
705   FORMAT(1H ,1XA3,3F6.0,2F7.0,F6.3,F7.2,F4.0,1PE10.3,0PF6.1)
      CALL WOUT(OTEMP,TL,RP3,R3,RPA3,TL3)
      TYPE 953,MODE,T,RP3,R3,RPA3,TL3
953   FORMAT(1H ,1XA3,F6.0,12X,2F7.3,4XF7.3,16XF6.1)
      IF(RSS.NE.'R') GO TO 760
      CALL SDR(XSA,YA,ARPM,MODER)
      TYPE 978,MODER,T,RN,PRS,TLRS
978   FORMAT(1H ,A4,F6.0,33X,2(1PE10.3),0PF6.1)
      CALL WOUT(OTEMP,TLRS,RP3,R3,RPA3,TL3)
      TYPE 954,MODER,T,TL3
954   FORMAT(1H ,A4,F6.0,53XF6.1)
C
760   CALL CZS(XSA,YA,THEB,THESD,RSD,RH,EXIST)
CCCC
C     TYPE 100,THEB,THESD,RSD,RH
100   FORMAT(' (CZS) THEB=',F7.2,', THESD=',F7.2,', RSD='F9.1,', RH=',
     1 F9.1)
      IF (EXIST.NE.'Y') GO TO 850
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DF,PHI,PRSQ,TL
      CALL WOUT(OTEMP,TL,RP3,R3,RPA3,TL3)
      TYPE 953,MODE,T,RP3,R3,RPA3,TL3
      IF (RSS.NE.'R') GO TO 850
      CALL CZSR(XSA,YA,THEB,THESD,RSD,RH,MODER)
      TYPE 978,MODER,T,RN,PRS,TLRS
      CALL WOUT(OTEMP,TLRS,RP3,R3,RPA3,TL3)
      TYPE 954,MODER,T,TL3
      GO TO 850
C
800   CALL CZ(XSA,YA,THEB,RTO,RH,EXIST)
      IF (EXIST.NE.'Y') GO TO 850
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DF,PHI,PRSQ,TL
      CALL WOUT(OTEMP,TL,RP3,R3,RPA3,TL3)
      TYPE 953,MODE,T,RP3,R3,RPA3,TL3
      IF (RSS.NE.'R') GO TO 850
      CALL CZR(XSA,YA,THEB,RTO,RH,MODER)
      TYPE 978,MODER,T,RN,PRS,TLRS
      CALL WOUT(OTEMP,TLRS,RP3,R3,RPA3,TL3)
```

```
      TYPE 954,MODER,T,TL3
C
850   CONTINUE
C
900   T=T+DT
      IF(T.LE.TF) GO TO 700
CCCC
C     TYPE 102,RR,RSD,RTO
102   FORMAT(' RR='F9.1,', RSD=',F9.1,', RTO='F9.1)
C
      M=M+1
      CALL WATIN(M,X3)
      IF (X3.EQ.13) GO TO 30
C
      END
C
C
      SUBROUTINE SD(XSA,YA,ARPM)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
C
      MODE='SD '
      CON=180./PI
      TA=T+(H/C1)+(RP/C2)
      TDIF=T-TO+(RP-RO)/C2
      PHI=CON*ACOS((-XSA)/RP)
      VA=SQRT(VAX*VAX+VAY*VAY)
      CPSI=(VAX*XSA-VAY*YA)/(VA*R)
      DOP=SQRT((1.0-2.0*VS*XSA/(C2*R))/(1.0-2.0*VA*CPSI/C2))
      DOE=SQRT(2.0*(CMAX-CS)/CMAX)
      DF=CON*DOE
      SG1=((CMAX-CMIN)/(DMIN-DS))**(1.0/3.0)
      SSM=9.0
      IF(WS.LT.10.0) SSM=6.75
      IF (WS.LT.7.0) SSM=4.5
      F1=F/1000.0
      ATOT=A+((7.4*FS)/((F1**(5.0/3.0))
     1 *SG1*DS*DS)+SSM*SQRT(F1/DS))/3000.0
      ARP=-ATOT*RP
      ARPM=0.0
      IF (ARP.GT.(-500.0)) ARPM=10.0**(ARP/10.0)
      PRSO=(((8.0/3.0)*AXN*AXN*(DOE**3))/(RP*DS))*ARPM
      TL=-999.0
      IF (PRSO.GE.ETA) TL=10.*ALOG10(PRSO)
C
CCCC
C     TYPE 100,A,ATOT,ARPM
100   FORMAT(' (SD) A='F8.3,', ATOT='F8.4,', NOT IN DB=',1PE10.4)
      RETURN
C
      END
C
      SUBROUTINE CZS(XSA,YA,THER,THESD,RSD,RR,EXIST)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
C
      MODE='CZS'
      CON=180.0/PI
      TA=T+(H/C1)+(RP/C2)
      TDIF=T-TO+(RP-RO)/C2
```

```
      PHI=CON*ACOS((-XSA)/RP)
      VA=SQRT(VAX*VAX+VAY*VAY)
      CPSI=(VAX*XSA-VAY*YA)/(VA*R)
      DOP=SQRT((1.0-2.0*VS*XSA/(C2*R))/(1.0-2.0*VA*CPSI/C2))
      GS=(CMAX-CS)/DS
      G1=(CMAX-CMIN)/(DMIN-DS)
      GFAC=(1.0/G1)+(1.0/GB)
      GFAC2=(1.0/GS)+(1.0/G1)
      RSD=2.0*CMAX*(GFAC*SQRT(1.0-CMIN*CMIN/(CMAX*CMAX))+
     1 SQRT(1.0-CS*CS/(CMAX*CMAX))/GS)
      RB=2.0*CB*(GFAC*SQRT(1.0-CMIN*CMIN/(CB*CB))+
     1 SQRT(1.0-CS*CS/(CB*CB))/GS-GFAC2*SQRT(1.0-CMAX*CMAX/(CB*CB)))
      IF ((RP.GE.RSD).AND.(RP.LE.RB)) GO TO 600
      IF ((RP.GE.RB).AND.(RP.LE.RSD)) GO TO 600
  500 EXIST='N'
      RETURN
C
  600 EXIST='Y'
      V=(VS-VAX)*(VS-VAX)+VAY*VAY
      XY=(XIS-XIA)*(XIS-XIA)+YIA*YIA
      ST=T/ABS(T)
      T1=TO+ST*SQRT(TO*TO-(XY-RB*RB)/V)
      T2=TO+ST*SQRT(TO*TO-(XY-RSD*RSD)/V)
      THESD=ACOS(CS/CMAX)
      DOE=THEB+(T-T1)*(THESD-THEB)/(T2-T1)
      DE=CON*DOE
      ARP=-A*RP
      ARPM=0.0
      IF (ARP.GT.(-500.0)) ARPM=10.0**(ARP/10.0)
      PRSQ=32.0*AXN*AXN*(THEB**3-THESD**3)*ARPM/
     1 (3.0*(THEB+THESD)*ABS(RB*RB-RSD*RSD))
      TL=-999.0
      IF (PRSQ.GE.ETA) TL=10.*ALOG10(PRSQ)
C
      RETURN
C
      END
C
      SUBROUTINE CZ(XSA,YA,THEB,RTO,RB,EXIST)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
C
      MODE='CZ '
      CON=180.0/PI
      TA=T+(H/C1)+(RP/C2)
      TDIF=T-TO-(RP-RO)/C2
      PHI=CON*ACOS((-XSA)/RP)
      VA=SQRT(VAX*VAX+VAY*VAY)
      CPSI=(VAX*XSA-VAY*YA)/(VA*R)
      DOP=SQRT((1.0-2.0*VS*XSA/(C2*R))/(1.0-2.0*VA*CPSI/C2))
      GS=(CS-CMIN)/DMIN
      GFAC=(1.0/GS)+(1.0/GB)
      RTO=2.0*CS*GFAC*SQRT(1.0-CMIN*CMIN/(CS*CS))
      RB=2.0*CB*(GFAC*SQRT(1.0-CMIN*CMIN/(CB*CB))-
     1 SQRT(1.0-CS*CS/(CB*CB))/GS)
      IF ((RP.GE.RTO).AND.(RP.LE.RB)) GO TO 600
      IF ((RP.GE.RB).AND.(RP.LE.RTO)) GO TO 600
  500 EXIST='N'
      RETURN
C
  600 EXIST='Y'
      V=(VS-VAX)*(VS-VAX)+VAY*VAY
      XY=(XIS-XIA)*(XIS-XIA)+YIA*YIA
```

```
      ST=T/ABS(T)
      T1=TO+ST*SQRT(TO*TO-(XY-RB*RB)/V)
      T2=TO+ST*SQRT(TO*TO-(XY-RTO*RTO)/V)
      DOF=THER*(1.0-(T-T1)/(T2-T1))
      DF=CON*DOF
      ARP=-A*RP
      ARPM=0.0
      IF (ARP.GT.(-500.0)) ARPM=10.0**(ARP/10.0)
      PRSO=32.0*AXN*AXN*THER*THER*ARPM/(3.0*ABS(RB*RB-RTO*RTO))
      TL=-999.0
      IF (PRSO.GE.ETA) TL=10.0*ALOG10(PRSO)
C
      RETURN
C
      END
C
      SUBROUTINE SDR(XSA,YA,ARPM,MODER)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,ES,GP,RO,TO,PI,
     1 GI
      COMMON/RS/ SW,CPW,PSIB,FPN,CI,PRS,TLRS,BN
C
      MODER=' SDR'
      CPW=(XWC*XSA-YWC*YA)/RP
      THE=DE*3.14159265/180.0
      CALL INT(THE)
      FC=1.E18
      IF (SW.NE.0.0) FC=1.0/(SQRT(2.0)*SW)
      CTA=COS(THE/2.0)
      STA=SIN(THE/2.0)
      ARG1=STA*FC/CTA
      ARG2=FC*(1.0-STA)/(CTA)
      CALL FRF(ARG1,FRF1)
      CALL FRF(ARG2,FRF2)
      BN=FPN*(FRF1+FRF2)
      PRS=(8.0*AXN*AXN*CI*BN*ARPM)/(RP*DS)
      TLRS=-999.0
      IF (PRS.GE.ETA) TLRS=10.0*ALOG10(PRS)
C
      RETURN
C
      END
C
      SUBROUTINE CZSR(XSA,YA,THER,THESD,RSD,RB,MODER)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,ES,GB,RO,TO,PI,
     1 GI
      COMMON/RS/ SW,CPW,PSIB,FPN,CI,PRS,TLRS,BN
C
      MODER='CZSR'
      CPW=(XWC*XSA-YWC*YA)/RP
      CALL INT(THER)
      CIB=CI
      CALL INT(THESD)
      FC=1.E18
      IF (SW.NE.0.0) FC=1.0/(SQRT(2.0)*SW)
      THS=THER+THESD
      CTS=COS(THS/2.0)
      STS=SIN(THS/2.0)
      ARG1=STS*FC/CTS
      ARG2=FC*(1.0-STS)/CTS
      CALL FRF(ARG1,FRF1)
```

```
      CALL FRF(ARG2,FRF2)
      RN=FPN*(FRF1*FRF2)
      ARPM=0.0
      APR=-A*RP
      IF(APR.GT.-500.0) ARPM=10.0**(APR/10.0)
      PRS=32.0*AXN*AXN*(CID-CI)*ARPM*RN/(THS*ABS(RP*RB-RSD*RSD))
      TLRS=-999.0
      IF (PRS.GE.ETA) TLRS=10.0*ALOG10(PRS)
      RETURN
      END
C
      SUBROUTINE CZR(XSA,YA,THER,RIO,RB,MODER)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DF,PHI,PRSD,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DP,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 GI
      COMMON/RS/ SW,CPW,PSIB,FPN,CI,PRS,TLRS,RN
C
      MODER=' CZR'
      CPW=(XWC*XSA-YWC*YA)/RB
      CALL INT(THER)
      FC=1.E18
      IF (SW.NE.0.0) FC=1.0/(SQRT(2.0)*SW)
      ARG=THER/2.0
      CT2=COS(ARG)
      ST2=SIN(ARG)
      ARG1=ST2*FC/CT2
      ARG2=FC*(1.0-ST2)/CT2
      CALL FRF(ARG1,FRF1)
      CALL FRF(ARG2,FRF2)
      RN=FPN*(FRF1*FRF2)
      ARPM=0.0
      APR=-A*RP
      IF(APR.GT.-500.0) ARPM=10.0**(APR/10.0)
      PRS=32.0*AXN*AXN*CI*RN*ARPM/(THER*ABS(RB*RB-RIO*RIO))
      TLRS=-999.0
      IF(PRS.GE.ETA) TLRS=10.0*ALOG10(PRS)
C
      RETURN
C
      END
C
      SUBROUTINE INT(TH)
C
      COMMON/RS/ SW,CPW,PSIB,FPN,CI,PRS,TLRS,RN
C
      IF (ABS(SW).GT.1.E-18) GO TO 12
      CI=((TH*PSIB*CPW)**3-(PSIB*CPW)**3)/3.0
C
      RETURN
C
12    SQPI=SQRT(3.14159265)
      SQ2=SQRT(2.0)
      TT=2.0/3.0
      FT=5.0/3.0
C
      PCS=PSIB*CPW/(SQ2*SW)
      PCS2=PCS*PCS
      TS=TH/(SQ2*SW)
      ARG=TS*PCS
      ARG2=ARG*ARG
      CALL FRF(ARG,FRFTP)
      CALL FRF(PCS,FRFP)
      CI=(((SQ2*SW)**3)/4.0)*(TS*TT*(ARG2*ARG-PCS2*PCS)+FRFTP*(1.0+TT
     1 *ARG2)*ARG-FRFP*(1.0+TT*PCS2)*PCS+EXP(-ARG2)*(FT+TT*ARG2)/SQPI
```

```
      2 -EXP(-PCS2)*(FT+TI*PCS2)/SQPI)
C
      RETURN
C
      END
C
      SUBROUTINE FRF(ARG,FRES)
C
      SARG=ARG/ABS(ARG)
      X=ABS(ARG)
      FR=1.0
      IF (X.GT.4.2) GO TO 200
      F=1.0+.0705230784*X+.0422820123*X*X+.0092705272*(X**3)+
      1 .0001520143*(X**4)+.0002765672*(X**5)+.0000430638*(X**6)
      FR=1.0-1.0/(F**16)
200   FRES=FR*SARG
C
      RETURN
C
      END
C
      SUBROUTINE WOUT(QTEMP,TLO,RP3,R3,RPA3,TL3)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
C
      RP3=RP/3000.0
      R3=R/3000.0
      X3=(VS*TA+XIS)-(VAX*TA+XIA)
      Y3= VAY*TA+YIA
      RPA3=SQRT(X3*X3+Y3*Y3)/3000.0
C
      TL3=-999.
      IF (TLO.EQ.-999.) RETURN
      TL9=10.0*ALOG10(9.0)
      CORR=(1.25E-6)*F*H
      IF((QTEMP.EQ.'Y').OR.(QTEMP.EQ.'YES'))CORR=(7.4E-8)*F*F*H/164.05
      TL3=TLO+TL9-CORR
C
      RETURN
C
      END
C
      SUBROUTINE WATIN(INM,K4)
C
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XWC,YWC,TI,TF,DT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,HO,IO,PI,
      1 GI
      COMMON/MM/M
C
      M=INM
      IF(M.GT.1) GO TO 3000
      TYPE 2000
2000  FORMAT(1H ,'TO CHANGE A RUN PARAMETER, ENTER THE'
      1 ' APPROPRIATE NUMBER:'/
      1 '    1  AIRCRAFT VEL. VECTORS'/
      1 '    2  AIRCRAFT INITIAL POSITION'/
      1 '    3  AIRCRAFT HEIGHT'/
      1 '    4  AIRCRAFT RADIATED FREQ.'/
      1 '    5  SUB VEL. VECTOR'/
      1 '    6  SUB INITIAL POSITION'/
      1 '    7  SUB DEPTH'/
      1 '    8  SURFACE SOUND SPEED'/
      1 '    9  BOTTOM SOUND SPEED AND DEPTH'/
      1 '    10 MINIMUM SOUND SPEED AND DEPTH'/
```

```
     1 '     11 MAX. SOUND SPEED AND DEPTH'/
     1 '     12 TIME OF EVENTS'/
     1 '     13 RUN'/
     1 '     14 STOP'/
     1 '     15 WIND PARAMETERS'/
     1 ' ENTER THE APPROPRIATE NUMBER = ',$)
       ACCEPT 2001,K3
2001   FORMAT(I)
       GO TO (47,48,49,10,11,12,13,14,15,16,19,20,30,18,984)K3
3000   TYPE 3001
3001   FORMAT(/' CHANGE PARAMETER = ',$)
       ACCEPT 3002,K4
3002   FORMAT(I)
       GO TO (47,48,49,10,11,12,13,14,15,16,19,20,30,18,984)K4
47     CALL AIRV(VAX,VAY)
       GO TO 3000
48     CALL AIRC(XIA,YIA)
       GO TO 3000
49     CALL AIRH(H)
       GO TO 3000
10     CALL AIRF(F)
       GO TO 3000
11     CALL SUBV(VS)
       GO TO 3000
12     CALL SUBC(XIS)
       GO TO 3000
13     CALL SUBD(D)
       GO TO 3000
14     CALL ENVS(CS)
       GO TO 3000
15     CALL ENVC(CB,DB)
       GO TO 3000
16     CALL ENVD(CMIN,DMIN)
       GO TO 3000
19     CALL ENVA(CMAX,DS)
       GO TO 3000
20     CALL TIM(TI,TF,DT)
       GO TO 3000
18     CALL EXIT
984    CALL WIND(WS,XWC,YWC)
       GO TO 3000
C
30     K3=13
       RETURN
C
       END
C
       SUBROUTINE WIND(WS,XWC,YWC)
       TYPE 200
200    FORMAT(1H ,'WIND SPEED=   ',$)
       ACCEPT 201,WS
201    FORMAT(F)
       TYPE 202
202    FORMAT('+X DIRECTION COSINE OF WIND=   ',$)
       ACCEPT 201,XWC
       TYPE 203
203    FORMAT('+Y DIRECTION COSINE OF WIND=   ',$)
       ACCEPT 201,YWC
       RETURN
       END
       SUBROUTINE AIRV(VAX,VAY)
       TYPE 200
200    FORMAT(1H ,'VELOCITY VECTOR X-DIRECTION(KTS)=',2X,$)
       ACCEPT 201,VAXKT
201    FORMAT(F)
       VAX=1.688*VAXKT
```

```
        TYPE 202
202     FORMAT('+VELOCITY VECTOR Y-DIRECTION(KTS)=',2X,$)
        ACCEPT 201,VAYKT
        VAY=1.688*VAYKT
        RETURN
        END
        SUBROUTINE AIRC(XIA,YIA)
        TYPE 200
200     FORMAT(1H ,'INTIAL X-COORDINATE OF AIRCRAFT( FT)=',2X,$)
        ACCEPT 201,XIA
201     FORMAT(F)
        TYPE 202
202     FORMAT('+INITIAL Y-COORDINATE OF AIRCRAFT(FT)=',2X,$)
        ACCEPT 201,YIA
        RETURN
        END
        SUBROUTINE AIRH(H)
        TYPE 200
200     FORMAT(1H ,'HEIGHT OF AIRCRAFT FROM SEA SURFACE(FT)=',2X,$)
        ACCEPT 201,H
201     FORMAT(F)
        RETURN
        END
        SUBROUTINE AIRF(F)
        TYPE 200
200     FORMAT(1H ,'FREQ. OF AIRCRAFT RADIATION(HZ)=',2X,$)
        ACCEPT 201,F
201     FORMAT(F)
        RETURN
        END
        SUBROUTINE SUBV(VS)
        TYPE 200
200     FORMAT(1H ,'VELOCITY VECTOR X-DIRECTION SUB(KTS)=',2X,$)
        ACCEPT 201,VSKT
201     FORMAT(F)
        VS=1.688*VSKT
        RETURN
        END
        SUBROUTINE SUBC(XIS)
        TYPE 200
200     FORMAT(1H ,'INITIAL X-COORDINATE OF SUB(FT)=',2X,$)
        ACCEPT 201,XIS
201     FORMAT(F)
        RETURN
        END
        SUBROUTINE SUBD(D)
        TYPE 200
200     FORMAT(1H ,'DEPTH OF SUB(FT)=',2X,$)
        ACCEPT 201,D
201     FORMAT(F)
        RETURN
        END
        SUBROUTINE ENVS(CS)
        TYPE 200
200     FORMAT(1H ,'SURFACE SOUND SPEED(FT/SEC)=',2X,$)
        ACCEPT 201,CS
201     FORMAT(F)
        RETURN
        END
        SUBROUTINE ENVC(CB,DB)
        TYPE 200
200     FORMAT(1H ,'BOTTOM SOUND SPEED(FT/SEC)=',2X,$)
        ACCEPT 201,CB
201     FORMAT(F)
        TYPE 202
202     FORMAT('+DEPTH FOR BOTTOM SOUND SPEED(FT)=',2X,$)
```

```
      ACCEPT 201,DH
      RETURN
      END
      SUBROUTINE ENVD(CMIN,DMIN)
      TYPE 200
200   FORMAT(1H ,'MIN. SOUND SPEED(FT/SEC)=',2X,$)
      ACCEPT 201,CMIN
201   FORMAT(F)
      TYPE 202
202   FORMAT('+DEPTH AT MIN. SOUND SPEED(FT)=',2X,$)
      ACCEPT 201,DMIN
      RETURN
      END
      SUBROUTINE ENVA(CMAX,DS)
      TYPE 200
200   FORMAT(1H ,'MAX. SOUND SPEED(FT/SEC)=',2X,$)
      ACCEPT 201,CMAX
201   FORMAT(F)
      TYPE 202
202   FORMAT('+DEPTH AT MAX. SOUND SPEED(FT)=',2X,$)
      ACCEPT 201,DS
      RETURN
      END
      SUBROUTINE TIM(TI,TF,DT)
      TYPE 200
200   FORMAT(1H ,'INITIAL TIME=',2X,$)
      ACCEPT 201,TI
201   FORMAT(F)
      TYPE 202
202   FORMAT('+FINAL TIME=',2X,$)
      ACCEPT 201,TF
      TYPE 203
203   FORMAT('+TIME INCREMENTS=',2X,$)
      ACCEPT 201,DT
      RETURN
      END
```

APPENDIX B

DIRECT PATH AND BOTTOM BOUNCE SUBROUTINE

Introduction

To calculate the transmission loss as a function of time for a moving source above the sea surface to a moving receiver below the sea surface through varying sea states (smooth to very rough), a computer program was developed. The sections which follow will be concerned with two modes of propagation from the source to receiver; direct and bottom bounce rays. The bottom bounce rays will be further subdivided into single bottom bounce or double bottom bounce rays.

Finally, two types of sound speed profiles, one with a surface duct present and one without, will be utilized.

To perform the calculations for the different modes of propagation involved, 15 separate subroutines are used with one main program as the control. Below is a listing of the main and subroutine programs used.

MAIN PROGRAM	SUBROUTINES	
WATER.F4	BOUN(THEθ)	BSSR(THEθ)
	SBOUN(THEθ)	BDSR(THEθ)
	BS(THEθ)	DP(THEθ)
	BD(THEθ)	DPR(THEθ)
	BSS(THEθ)	ERFS.F4
	BDS(THEθ)	BT3.F4
	BSR(THEθ)	BT5.F4
	BDR(THEθ)	

Figure E-1 is a block diagram showing the flow of the calculations. This chart gives an overall picture of the flow of the program calculations during the actual execution.

B.1 Brief Description and Purpose of Program

MAIN PROGRAM - WATER.F4

The main program has several functions. One of these is accepting all the input parameters which initialize the problem. The input parameters are divided into four main categories: (1) source parameters, (2) receiver parameters, (3) environmental parameters, and (4) time parameters. Once all the parameters are inputted, a series of calculations are performed to obtain a set of constants for each particular time. These are stored in a common block with all the input parameters so that they can be shared by all the subroutines. The third function of the main program is to route the flow through 4 main loops by a series of conditional statements. Finally, the main program controls the output of values from each individual subroutine so that the values are displayed in a pre-assigned format.

BOUN(THEθ)

This subroutine calculates the angle of the arriving ray which reaches the receiver with no surface duct present. It first determines whether there is a single or double bounce which reaches the receiver. An iteration routine is then used to finally calculate the acquired ray if there is one. (NOTE: if no single bounce exists, the program defaults to double bounce; if the double bounce does not exist, the control is sent back to main program where DP.F4 is called.) Now depending on whether the solution is a single or double bounce, the main control is either sent to BS(THEθ) or BD(THEθ).

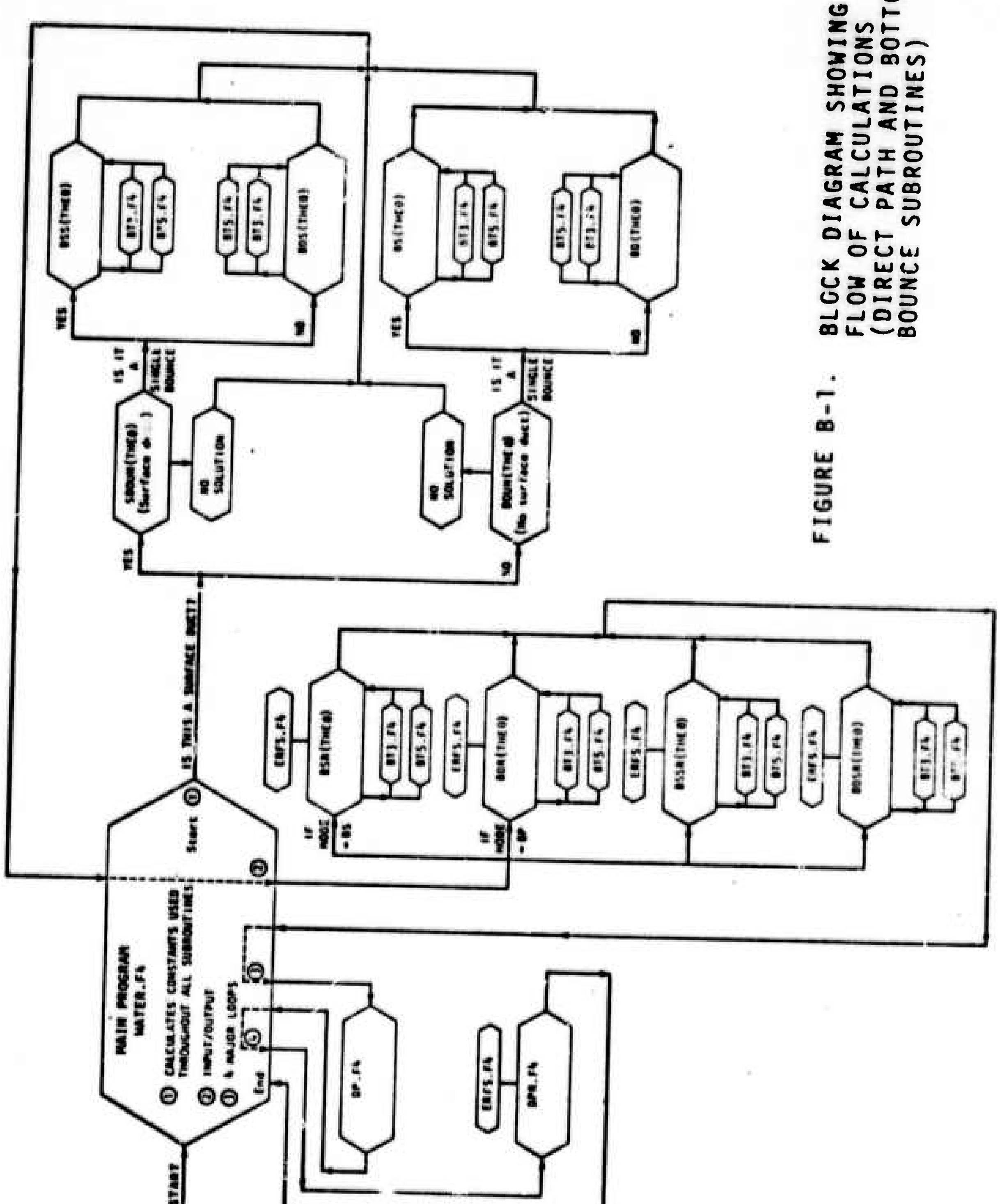


FIGURE B-1. BLOCK DIAGRAM SHOWING FLOW OF CALCULATIONS (DIRECT PATH AND BOTTOM BOUNCE SUBROUTINES)

SBOUN(THEθ)

The difference between SBOUN(THEθ) and BOUN(THEθ) is that a surface duct is present for the calculation of the angle of the arriving ray which reaches the receiver. Again, it also has to determine first whether there is a single or double bounce which reaches the receiver with the surface duct present. An iteration routine is then used to finally calculate the acquired ray, if there is one. Depending on whether the solution is a single or double bounce, the main control is sent either to BSS(THEθ) or BDS(THEθ).

BS(THEθ)

The single bounce rays with no surface duct present, are handled within this subroutine. Besides calculations to obtain the TL for this particular mode of propagation, other important quantities of interest are also calculated. The final output values from this subroutine are sent to the main program where they are printed out using the main program format.

BD(THEθ)

This is the double bounce subroutine which performs all the calculations for double bounce rays with no surface duct present. Many of the formula are modified versions of the single bounce equations. The final output values from this subroutine are again sent to the main program where they are printed out.

BSS(THEθ)

For bottom bounce rays which are in the presence of a surface duct, the calculations are performed within this subroutine. The final output values are returned to the main program where they are printed out.

BDS(THEθ)

This is the subroutine which calculates double bounce rays propagation with a surface duct present.

BSR(THEθ)

All subroutines with the letter R contained in the name are used for rough surface calculations. This subroutine modifies the smooth surface results for the single bounce mode of propagation. It has the necessary equations to calculate TL for a rough sea surface. Once all the output values are calculated, they are sent to the main control program to be printed out. No surface duct is present.

BDR(THEθ)

This is the subroutine which handles rough surface double bounce propagation (no surface duct present). Again it returns to main program with a new PRATIO and TL for this mode of propagation in a rough sea.

BSSR(THEθ)

To take into account a surface duct in a rough sea, subroutine BSSR(THEθ) handles the single bottom bounce mode of propagation for this condition. It calculates a new PRATIO and TL due to the change in sea surface conditions. It returns to main program with output values.

BDSR(THEθ)

This subroutine calculates the double bounce ray for a rough sea surface and surface duct present. It modifies the smooth surface formula and calculates a new PRATIO and TL due to the rough sea state. It returns to main program with output values.

DP(THEθ)

For direct path propagation all calculations are handled within this subroutine for smooth surface conditions.

DPR(THEθ)

The direct path propagation TL with a rough sea surface are handled by the formula which are modified to take into account the rough sea surface.

ERFS.F4

To obtain the statistics for the rough surface subroutines, the error function is necessary. This subroutine calculates the error function for a given input.

BT3.F4 and BT5.F4

These two subroutines are used to obtain the bottom reflection loss as a function of frequency and grazing angle. Bottom types 3 or 5 are typical of most of the ocean bottoms of the world. This data was obtained from the FACT model.

B.2 Instructions for Execution

To execute the calculations, 6 programs must be loaded. They are WATER.F4, BO.F4, SBO.F4, BT3.F4, BT5.F4, and ERFS.F4. Once loaded, the main program starts the control which asks the user to input all the parameters necessary for the calculation. The program asks for the following inputs listed below: (The inputs must be floating point numbers.)

INPUT	UNITS	COMPUTER PGM NAME
Aircraft velocity x-direction	knots	VAX
Aircraft velocity y-direction	knots	VAY
Initial x-coordinate of aircraft (at t=0)	feet	XIA
Initial y-coordinate of aircraft (at t=0)	feet	YIA
Height of aircraft from sea surface	feet	H
Frequency of aircraft radiation	Hz	F
Receiver velocity x-direction	knots	VS
Initial x-coordinate of receiver (at t=0)	feet	XIS
Depth of receiver	feet	D
Wind speed	knots	V
x-direction cosine of wind direction	dimensionless	WSX
y-direction cosine of wind direction	dimensionless	WSY
Bottom type (3 or 5)	dimensionless	BT
Starting time	seconds	TI
Final time	seconds	TF
Time increments	seconds	DT
Is there a surface duct	yes/no	
Surface sound speed	ft/sec	CS
Bottom sound speed	ft/sec	CB
Depth for bottom sound speed	ft	DB
Minimum sound speed	ft/sec	CMIN
Depth of minimum sound speed	ft	DMIN
Maximum sound speed	ft/sec	CMAX
Depth at maximum sound speed	ft	DS

The output is formatted such that a heading is printed first with the following names: MODE, T, TA, TDIF, R', R, DOP, D/E, PHI, PRATIO, TL.

MODE - Mode of propagation

T - Start time of ray

TA - Time of arrival of ray

TDIF - difference between TA and time of arrival of direct path ray from CPA point

R' - Lateral range between source and receiver

R - Slant range between source and receiver

DOP - Doppler

D/E - Arrival angle at receiver

PHI - Azimuthal arrival angle

PRATIO - Received mean square pressure

TL - Transmission loss between source and receiver

The corresponding value for each heading is given in a column below the headings.

Once the set of calculations have been performed the program prints out a message to ask the user whether he would like to run the program again with any of the input parameters changed. The user may opt to run with a new set of parameters or stop the execution totally by typing the pre-assigned number.

The following pages are a listing of a sample run showing how the input parameters are given and the actual formatted output.

```
.EX WATER.F4,30.F4,S30.F4,3T3.F4,TT5.F4,ERFS.F4
FORTRAN:  ERFS.F4
LOADING

LOADER 11K CORE
13+3K MAX 478 WORDS FREE
EXECUTION
```

INPUT
PARAMETERS

```
ENTER THE APPROPRIATE PARAMETERS IN
THE DIMENSIONS INDICATED.

VELOCITY VECTOR X-DIRECTION(KTS)=  220.0

VELOCITY VECTOR Y-DIRECTION(KTS)=  0.0

INTIAL X-COORDINATE OF AIRCRAFT( FT)=  0.0*

INITIAL Y-COORDINATE OF AIRCRAFT(FT)=  1000.0*

HEIGHT OF AIRCRAFT FROM SEA SORFACE(FT)=  50.0

FREQ. OF AIRCRAFT RADIATION(HZ)=  1000.0

VELOCITY VECTOR X-DIRECTION SUB(KTS)=  10.0

INITIAL X-COORDINATE OF SUB(FT)=  0.0*

DEPTH OF SUB(FT)=  200.0

WIND SPEED(KTS)=  24.5

X-DIRECTION COSINE=  1.0

Y-DIRECTION COSINE=  0.0

BOTTOM TYPE EITHER 3.0 OR 5.0=  3.0

INITIAL TIME(sec)=  1100.0†

FINAL TIME(sec)=  0.0

TIME INCREMENTS(sec)=  1.0

IS THERE A SURFACE DUCT  YES

SURFACE SOUND SPEED(FT/SEC)=  4975.0

BOTTOM SOUND SPEED(FT/SEC)=  5100.0

DEPTH FOR BOTTOM SOUND SPEED(FT)=  18000.0

MIN. SOUND SPEED(FT/SEC)=  4887.0

DEPTH AT MIN. SOUND SPEED(FT)=  2300.0

MAX. SOUND SPEED(FT/SEC)=  4997.0

DEPTH AT MAX. SOUND SPEED(FT)=  1200.0
```

* Here, "initial" refers to t=0

† In this statement, "initial" refers to the starting time of the encounter.

```
TRYING A SURFACE DUCT WITH DOUBLE BOUNCE SOLUTION

 0.324E+01 0.300E+01             OUTPUT PARAMETERS
THEO= 0.131E+02
MODE     T     TA   TDIF      R'        R   DOP    D/E   PHI    P RATIO     TL
 SDS   1100.  1180.  1179.389929.389929.0.929  13.11   0.  4.220E-14-147.7
0.3544E+00
0.3838E+00
0.1184E-01                                  EXTRANEOUS:  ONLY FOR
0.6292E-01                                  CHECKING MECHANICS OF
0.1877E-06                                  PROGRAM
0.6603E+00
0.5217E+00
0.3674E-01
0.3241E+010.3000E+01
0.1997E+06
 BDSR 1100.  1180.  1179.389929.389929.0.929  13.11   0.  1.212E-15-149.2
 DP   1100.  1178.  1178.389929.389929.0.936   0.03   0.  3.350E-19-1[illegible]2.5
0.3544E+00
0.3838E+00
0.1184E-01
0.6292E-01
0.1877E-06
0.6603E+00
0.5217E+00
 DPR  1100.  1178.  1178.389929.389929.0.936   0.03   0.  2.696E-15-145.7
TO CHANGE A RUN PARAMETER, ENTER THE
APPROPRIATE NUMBER:
  1  AIRCRAFT VEL. VECTORS
  2  AIRCRAFT INITIAL POSITION
  3  AIRCRAFT HEIGHT
  4  AIRCRAFT RADIATED FREQ.
  5  SUB VEL. VECTOR
  6  SUB INITIAL POSITION
  7  SUB DEPTH
  8  SURFACE SOUND SPEED
  9  BOTTOM SOUND SPEED AND DEPTH
 10  MINIMUM SOUND SPEED AND DEPTH
 11  WIND SPEED AND DIRECTION COSINES
 12  BOTTOM TYPE
 13  MAX. SOUND SPEED AND DEPTH
 14  TIME OF EVENTS
 15  RUN
 16  STOP
ENTER THE APPROPRIATE NUMBER = 16

EXIT

.KJOB
USER [217,11] JOB 27 ISC#2 OFF TTY15 AT 10:24 AM THU 18-DEC-75
CONNECT TIME 0:12  CRU'S 354
```

B.3 Example of Program Flow

For the user to understand the program flow, an actual case will be simulated. This way the user may follow through to see which subroutines and formula are used for one specific case.

As shown in the section on execution of the program, all the inputs are manually inserted first. Once this is done the main program calculates a series of constants set up by the main program internally. These are constants which do not change with time. A listing of these constants are given below:

Constant	Computer Program Name
GS = (CS-CMIN)/DMIN	GS (surface gradient)
GB = (CB-CMIN)/(DB-DMIN)	GB (bottom gradient)
C1 = 1100	C1 (sound speed in air, in ft/sec)
AXN = C1/C2	AXN (ratio of air sound speed and water sound speed)
C2 = 5000	C2 (average sound speed in water in ft/sec)
$t_0 = \frac{(XIA-XIS)(VS-VAX)-(YIA)(VAY)}{[(VS-VAX)^2+VAY^2]}$	TO time of closest approach
$R(t_0) = [(X_s(t_0)-X_a(t_0))^2+Y_a^2(t_0)+D^2]^{1/2}$	RO slant range at closest approach
$A = \left[\frac{.1(F_{kHz})^2}{1+F_{kHz}^2}+\frac{40(F_{kHz})^2}{4100+F_{kHz}^2}\right]\frac{1}{3000}$	A volumetric atten (DB/ft)

$F_{kHz} = F/1000$

$R'(t) = [\{X_s(t) - X_a(t)\}^2 + Y_a^2(t)]^{\frac{1}{2}}$ RP Lateral Range

$R(t) = [\{X_s(t) - X_a(t)\}^2 + Y_a^2(t) + D^2]^{\frac{1}{2}}$ R slant range

If a surface duct present then:

G1 = (CMAX-CMIN)/(DMIN-DS) G1

GS = (CMAX-CS)/DS GS

With all of the necessary parameters inputted and all the constants initialized, the main program sets up a common storage area so that all this data may be shared with all other subroutines.

Now with all the necessary parameters initialized and set up so that they can be shared by all subroutines, the main program now tests for whether there is a surface duct or not. For our case there is no surface duct so the subroutine BOUN(THEθ) is called. This subroutine calculates the arrival angle of the ray which reaches the receiver. At a time instant t, the subroutine first tests whether there is a single or double bounce and this is done by testing whether the distance to the convergence zone is less than or greater than the lateral distance between the receiver and the virtual source.

$$R_b = 2c_b\left[\frac{1}{g_s}\left(\sqrt{1-\left(\frac{c_{min}}{c_b}\right)^2} - \sqrt{1-\left(\frac{c_s}{c_b}\right)^2}\right) + \frac{1}{g_b}\sqrt{1-\left(\frac{c_{min}}{c_b}\right)^2}\right]$$

Lateral receiver/virtual source range at t

$$B(\theta) = \left[\{X_a(t-h/c_1) - X_s(t)\}^2 + y_a^2(t-h/c_1)\right]^{1/2}$$

If $B(\theta) > R_b$ then it is a double bounce, if not it is a single bounce. Now an iteration routine is used to finally calculate the acquired ray angle. Once this is obtained* and for purposes of simulation a single bounce solution will be used so that subroutine BS(THEθ) is called. The following calculations are performed within BS(THEθ).

Bounce distance: $$B(\theta_0) = \frac{2c_s}{\cos\theta_0}\left[\frac{1}{g_s}\left(\sqrt{1-\left(\frac{c_{min}}{c_s}\right)^2\cos^2\theta_0} - \sin\theta_0\right) + \frac{1}{g_b}\left(\sqrt{1-\left(\frac{c_{min}}{c_s}\right)^2\cos^2\theta_0} - \sqrt{1-\left(\frac{c_b}{c_s}\right)^2\cos^2\theta_0}\right)\right]$$

Travel Time Along BB Ray: $$t(\theta_0) = \frac{2}{g_s}\left[\cosh^{-1}\left(\frac{c_s}{c_{min}\cos\theta_0}\right) - \cosh^{-1}\left(\frac{1}{\cos\theta_0}\right)\right] - \frac{2}{g_b}\left[\cosh^{-1}\left(\frac{c_s}{c_b\cos\theta_0}\right) - \cosh^{-1}\left(\frac{c_s}{c_{min}\cos\theta_0}\right)\right]$$

ARRIVAL ANGLE: D/E = THEO

$$\frac{\partial B(\theta_0)}{\partial\theta_0} = B(\theta_0)\tan\theta_0 + 2c_s\left[\frac{1}{g_s}\left\{\frac{\left(\frac{c_{min}}{c_s}\right)^2\sin\theta_0}{\sqrt{1-\left(\frac{c_{min}}{c_s}\right)^2\cos^2\theta_0}} - 1\right\} + \frac{\sin\theta_0}{g_b}\left\{\frac{\left(\frac{c_{min}}{c_s}\right)^2}{\sqrt{1-\left(\frac{c_{min}}{c_s}\right)^2\cos^2\theta_0}} - \frac{(c_b/c_s)^2}{\sqrt{1-\left(\frac{c_b}{c_s}\right)^2\cos^2\theta_0}}\right\}\right]$$

---

*If no solution for either single or double bounce, the control is sent back to main program where DP.F4 is called.

PATH LENGTH: $S(\theta_0) = \dfrac{2c_s}{\cos\theta_0}\left[-\left\{\dfrac{1}{g_s}+\dfrac{1}{g_b}\right\}\sin^{-1}\left(\dfrac{c_{min}}{c_s}\cos\theta_0\right)+\dfrac{1}{g_b}\sin^{-1}\left(\dfrac{c_b}{c_s}\cos\theta_0\right)\right.$

$\left.+\dfrac{1}{g_s}\left(\dfrac{\pi}{2}-\theta_0\right)\right]$

AZIMUTHAL ARRIVAL ANGLE: $\cos\phi(t_A) = \dfrac{\{X_a(t-h/c_1) - X_s(t+t(\theta_0))\}}{\left[\{X_a(t-\frac{h}{c_1}) - X_s(t+t(\theta_0))\}^2 + y_a^2(t-\frac{h}{c_1})\right]^{1/2}}$

AUXILIARY ANGLE FOR DOPPLER:

$$\cos\psi = \frac{v_{ax}\left[X_s(t+t(\theta_0)) - X_a(t-\frac{h}{c_1})\right] + v_{ay}\left[-y_a(t-h/c_1)\right]}{|v_a|\left[\{X_s(t+t(\theta_0)) - X_a(t-\frac{h}{c_1})\}^2 + y_a^2(t-\frac{h}{c_1}) + (2D_B-D)^2\right]^{1/2}}$$

DOPPLER SHIFT:

$$\frac{f_R}{f_s} = \frac{1}{\left[1+\frac{v_a^2}{c_2^2} - 2\frac{|v_a|}{c_2}\cos\psi\right]^{1/2}}\left[1+\frac{v_s^2}{c_2^2} - \frac{2v_s\{X_s(t+t(\theta_0)) - X_a(t-h/c_1)\}/c_2}{\left[\{X_s(t+t(\theta_0)) - X_a(t-\frac{h}{c_1})\}^2 + y_a^2(t-\frac{h}{c_1}) + (2D_B-D)^2\right]^{1/2}}\right]^{1/2}$$

TIME OF ARRIVAL MINUS TIME OF ARRIVAL OF DIRECT PATH FROM CPA POINT:

$$t_A - t_{AMAX} = t + t(\theta_0) - t_0 - \frac{R(t_0)}{c_2} - \frac{h}{c_1}$$

At this point, the angle of the grazing ray to the bottom is calculated and depending on which bottom type (3 or 5) is inputted, subroutine BT3.F4 or BT5.F4 is called with the grazing ray angle and returns with a bottom loss. The subroutine BT3.F4 or BT5.F4 has a series of linear equations which are good for specific frequency and grazing angle ranges. The subroutine interpolates to find the best fit. The bottom loss values were programmed using LRAPP data.

It is now possible by combining several of the parameters calculated already to obtain the mean square pressure at the receiver and the TL.

$$\frac{P_{BB}^2}{P_a^2} = \frac{8\,m^2 \sin\theta_o \cos\theta_o}{B(\theta_o)\left|\frac{\partial B(\theta_o)}{\partial \theta_o}\right|}\; 10^{-\{AS(\theta_o) + RBL(\theta_b)\}/10}$$

$$TL = 10 \log_{10}\left\{\frac{8\,m^2 \sin\theta_o \cos\theta_o}{B(\theta_o)\left|\frac{\partial B(\theta_o)}{\partial \theta_o}\right|}\right\} - AS(\theta_o) - RBL(\theta_b)$$

A correction for air attentuation is added to the TL.*

The control is now sent back to the main program where the values of MODE, T, TA, TDIF, R', R, DOP, D/E, PHI, PRATIO, TL are printed in a preassigned format.

The main program now goes into a series of conditional statements to find out which mode was just calculated for the smooth sea surface case, (which in this case the mode = BS) and routes the program to call the subroutine for the rough surface case which for BS is BSR(THEθ).

---

* See Eq. 64, Appendix A.

Once within this subroutine a series of rough surface formula to set up the statistics for the rough surface case are calculated as given below:

Azimuthal angle between plane of acoustic path and wind direction at arrival time t_a

$$\cos\phi_w(t_a) = \frac{m_{sx}\{X_s(t+t(\theta_0)) - X_a(t-h/c_1)\} - m_{sy}\, y_a(t-h/c_1)}{\left[\{X_s(t+t(\theta_0)) - X_a(t-\frac{h}{c_1})\}^2 + y_a^2(t-\frac{h}{c_1})\right]^{1/2}}$$

AUXILIARY VARIABLE:

$$\xi = \left\{\frac{8h\lambda_{AIR}\, g^2}{\pi^2 U^4}\right\}^{1/4}$$

Having a value of zeta (ξ), the subroutine now calls a separate subroutine ERFS( ) to obtain a value of the error function for that particular value of ξ. Once the value is obtained it is returned to the calling subroutine.

Mean square sea slope (fully arisen sea)

$$\sigma^2 = 1.15\times10^4 \sqrt{\frac{\pi}{2}}\,\frac{U}{g^3}\,[1-\mathrm{erf}(\xi)]$$

rms slope in wind direction

$$\sigma_w = \sigma/\sqrt{2}$$

Most probable slope estimate

$$\overline{\psi} = \frac{2.86}{1000} U\left(\frac{cm}{sec}\right) \frac{\pi}{180}$$

Mean square sea surface curvature:

$$\overline{\eta^2} = 2.3\times10^4 \frac{\sqrt{2}}{gU^3} \frac{4}{3}\left[\sqrt{\pi}\left\{1-erf(\xi)\right\} + \frac{e^{-\xi^2}}{\xi^2}\left(\frac{1}{2\xi^2}-1\right)\right]$$

Auxiliary variable:

$$m = h\sqrt{\overline{\eta^2}}$$

Again the subroutine erfs( ) is called and values of the error function are obtained for

$$\frac{1}{m}, \quad \frac{\tan\theta}{\sqrt{2}\,\sigma_w}, \quad \left[\frac{\frac{1-\sin\theta}{\cos\theta}}{\sqrt{2}\,\sigma_w}\right]$$

Once these values are obtained, the equation for the average number of refracting paths (fully arisen sea) can be calculated:

$$\bar{N} = \left\{\frac{m^2}{4\pi}\left[e^{-1/m^2} + \frac{\sqrt{\pi}}{2m}\left\{1+erf\left(\frac{1}{m}\right)\right\}\right]^2 2\right\}\left[erf\left(\frac{\tan\theta}{\sqrt{2}\,\sigma_w}\right) + erf\left(\frac{\left\{\frac{1-\sin\theta}{\cos\theta}\right\}}{\sqrt{2}\,\sigma_w}\right)\right]$$

For the rough surface case only the mode, the mean square pressure and the TL are modified from the smooth surface case; all other output is exactly the same. The new modified mean square pressure and TL are given in the equations below:

$$\frac{P_{BBR}^2}{P_a^2} = \frac{8m^2\cos\theta_0}{\sin\theta_0\, B(\theta_0)\left|\frac{\partial B(\theta_0)}{\partial\theta_0}\right|}\left[\frac{1}{2}\left\{1+\operatorname{erf}\left(\frac{\tan\theta_0+\bar{\Psi}\cos\phi_w}{\sqrt{2}\,\sigma_w}\right)\right\}\cos^2\theta_0\left[\sigma_w^2+\right.\right.$$

$$\left.\left.(\tan\theta_0+\bar{\Psi}\cos\phi_w)^2\right]+\frac{\sigma_w}{\sqrt{2\pi}}\cos^2\theta_0\,(\tan\theta_0+\bar{\Psi}\cos\phi_w)\,e^{-\frac{(\tan\theta_0+\bar{\Psi}\cos\phi_w)^2}{2\sigma_w^2}}\right]$$

$$\times\ \bar{N}\ 10^{-(AS(\theta_0)+RBL(\theta_b))/10}$$

$$TL = 10\,\mathrm{Log}_{10}\left[\frac{P_{BBR}^2}{P_a^2}\right]$$

With the two new values the control is referred back to the main program where the output values are again printed of MODE, T, TA, TDIF, R', R, DOP, D/E, PHI, PRATIO, and TL.

The main control program now automatically goes to the subroutine DP.F4 to calculate for direct path propagation. Within DP.F4 the following equations are solved.

Depression angle

$$\sin\theta(t) = \frac{D}{\left[\{X_s(t)-X_a(t)\}^2+y_a^2(t)+D^2\right]^{1/2}}$$

Azimuthal arrival angle

$$\cos\phi(t) = \frac{\{X_a(t) - X_s(t)\}}{\left[\{X_a(t) - X_s(t)\}^2 + y_a^2(t)\right]^{1/2}}$$

Lateral range

$$R'(t) = \left[\{X_s(t) - X_a(t)\}^2 + y_a^2(t)\right]^{1/2}$$

Slant range

$$R(t) = \left[\{X_s(t) - X_a(t)\}^2 + y_a^2(t) + D^2\right]^{1/2}$$

Time of closest approach

$$t_o = \frac{(X_{ia} - X_{is})(v_s - v_{ax}) - y_{ia} v_{ay}}{\left[(v_s - v_{ax})^2 + v_{ay}^2\right]}$$

Time difference between arrival time and arrival time of direct path from CPA point

$$\Delta t = t - t_o + \frac{R(t) - R(t_o)}{c_2}$$

Time of arrival

$$t_A = t + \frac{h}{c_1} + \frac{R(t)}{c_2}$$

Auxiliary variable:

$$\cos\psi(t) = \frac{v_{ax}\{X_s(t) - X_a(t)\} - v_{ay} y_a(t)}{v_a R(t)}$$

Doppler:

$$\frac{f_R}{f_s} = \left[1 - 2\frac{v_a}{c_2}\cos\psi(t)\right]^{-1/2} \left[1 - \frac{2v_s}{c_2}\left\{\frac{X_s(t) - X_a(t)}{R(t)}\right\}\right]^{1/2}$$

Squared Pressure:

$$\frac{P^2_{DP}(t)}{P^2_a} = \left(\frac{2mD}{\left[\{X_s(t) - X_a(t)\}^2 + y_a^2(t) + D^2\right]}\right)^2$$

Transmission Loss:

$$TL = 10\,\mathrm{Log}_{10}\left[\frac{P^2_{DP}(t)}{P^2_a}\right] - AR(t)$$

The control is now sent to the main program where the values outputted are printed MODE, T, TA, TDIF, R', R, DOP, D/E, PHI, PRATIO, and TL. Now the final main loop is entered which is the calling of the subroutine DPR.F4 which does the calculations for the rough surface case of direct path propagation.

The initial rough surface formula for the statistics of the rough sea surface are the same; that is

$$\xi\,,\ \sigma^2,\ \sigma_w\,,\ \overline{\Psi}\,,\ \overline{\eta^2}\,,\ m\,,\ \overline{N}$$

are the same exact equations only

$$\cos\phi_w(t_A)$$

is different as shown below

$$\cos\phi_w(t_A) = \frac{n_{sx}\{X_s(t) - X_a(t)\} - n_{sy}\,y_a(t)}{\sqrt{\{X_s(t) - X_a(t)\}^2 + y_a^2(t)}}$$

Therefore the modified mean square pressure and TL for the direct path rough surface case are:

$$\frac{P_{DP}^2}{P_a^2} = \left\{\frac{2m}{R(t)}\right\}^2 \left[\frac{1}{2}\left\{1+\operatorname{erf}\left(\frac{\tan\theta+\bar{\Psi}\cos\phi_w}{\sqrt{2}\,\sigma_w}\right)\right\}\cos^2\theta\left[\sigma_w^2+(\tan\theta+\bar{\Psi}\cos\phi_w)^2\right]\right.$$

$$\left.+\frac{\sigma_w}{\sqrt{2\pi}}\cos^2\theta\,(\tan\theta+\bar{\Psi}\cos\phi_w)\,e^{-\frac{(\tan\theta+\bar{\Psi}\cos\phi_w)^2}{2\sigma_w^2}}\right]\cdot\bar{N}\cdot 10^{-AR(t)/10}$$

$$TL = 10\log_{10}\left[\frac{P_{DP}^2}{P_a^2}\right]$$

A correction for air attenuation is then added to the TL.*

The final values are returned to the main program where they are printed out in the specified format, the values printed at MODE, T, TA, TDIF, R', R, DOP, D/E PHI, PRATIO, TL.

The main control program has now completed all of the necessary data reduction; it then asks the user if he would like to change anything and rerun the program; if not, the user can exit and end the calculations.

---

* See Eq. 64 of Appendix A

B.4 Output Description

The output of the programs differs slightly depending upon (1) whether a surface duct is present or not and (2) if the program is outputting single or double bounce results. Examples of the output for each of the four possible cases are given and described.

CASE 1, OUTPUT (A): NO SURFACE DUCT, DOUBLE BOUNCE OUTPUT

```
TRYING A DOUBLE BOUNCE SOLUTION
(15)0.525E+01 (16)0.000E+00
THE0= 0.537E+01 (17)
MODE      T      TA    TDIF        R'       R   DOP     D/E   PHI   P RATIO      TL
 BD    -900.   -832.   -846.324478.324479.1.076     5.37 176.  8.353E-14-130.8
                            108.159 100.044                      -121.3
  0.5962E+01 (18)
  0.1000E+01 (19)
  0.0000E+00 (20)
  0.2260E-01 (21)
  -.2864E-22 (22)
  0.1631E-05 (23)
  0.5000E+00 (24)
(25)0.5248E+01 0.0000E+00 (26)  (5)     (6)     (7)         (9)      (10)      (11)
(1)BDR (2)-900. (3)-832. (4)-846.324478.324479.1.076 (8) 5.37 176.  1.116E-13-129.5
                       (13)108.159 100.044(12)              (14)  -120.1
 DP    -900.   -826.   -840.324478.324479.1.081     0.07176.  2.631E-18-175.8
                            108.159 99.319                        -166.4
 0.5962E+01 (18)
 0.1000E+01 (19)
 0.0000E+00 (20)
 0.2260E-01 (21)
 -.2864E-22 (22)
 0.1631E-05 (23)
 0.5000E+00 (27)
 DPR   -900.   -826.   -840.324478.324479.1.081     0.07176.  4.424E-16-153.5
                            108.159 99.319                        -144.1
```

For each particular mode of propagation, two main lines of output are printed. Numbers 1-14 are the main output values. They are:

(1) Mode of propagation

(2) Running time parameter t

(3) Time of arrival (secs) t_A

(4) Difference in time between arrival time of mode t_A and time of arrival of direct path from CPA (secs)

(5) Lateral range between source and receiver (ft) at time signal leaves source (t)

(6) Slant range between source and receiver (ft) at time signal leaves source (t)

(7) Doppler shift at arrival time t_A

(8) D/E arrival angle at receiver (deg) at t_A

(9) Azimuthal arrival angle (deg) at t_A

(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses) at t_A

(11) Transmission loss re 1 ft (dB) at t_A

(12) Lateral range (kiloyards), at time signal leaves source (t)

(13) Lateral range at time of arrival (kiloyards) at t_A

(14) Transmission loss re 1 yd (dB), includes air attenuation, at t_A

Also outputted are some supporting data used as checks, they are; (15-27):

(15) Grazing angle at bottom (deg)

(16) Bottom loss from LRAPP data for this grazing angle

(17) D/E angle at receiver

(18) $\xi = (8h\lambda_{air} g^2/\pi^2 U^4)^{1/4}$

(19) ERF(ξ)

(20) σ^2 (mean square sea slope (fully arisen sea))

(21) Most probable slope estimate (radians)

(22) $\overline{\eta^2} = 2.3\times10^4 \frac{\sqrt{2}}{gU^3} \frac{4}{3}\left[\sqrt{\pi}\{1-erf(\xi)\} + \frac{e^{-\xi^2}}{\xi}\left\{\frac{1}{2\xi^2}-1\right\}\right]$

(23) $m = h\sqrt{\overline{\eta^2}}$

(24) First part of $\bar{N}$ (Average number of refracting paths)

$$= \frac{m^2}{4\pi}\left[e^{-1/m^2} + \frac{\sqrt{\pi}}{2m}\left\{1+\mathrm{erf}\left(\frac{1}{m}\right)\right\}\right]^2 2$$

(25) Grazing angle at bottom (deg) (same as (15))

(26) Bottom loss from LRAPP (same as (16))

(27) Same as (24)

CASE 1, OUTPUT B: NO SURFACE DUCT — SINGLE BOUNCE OUTPUT

```
THERE IS A SINGLE BOUNCE

 0.582E+00 0.000E+00
THEO= 0.128E+01
MODE     T      TA   TDIF      R'        R   DOP    D/E   PHI     P RATIO     TL
 BS    -600.  -555.  -569.217057.217058.1.076    1.28174. 2.258E-14-136.5
                           72.352 66.967                          -127.0
0.5962E+01
0.1000E+01
0.0000E+00
0.2260E-01
-.2864E-22
0.1631E-05
0.5000E+00
0.5821E+000.0000E+00
 BSR   -600.  -555.  -569.217057.217058.1.076    1.28174. 5.983E-14-132.2
                           72.352 66.967                          -122.8
 DP    -600.  -547.  -561.217057.217058.1.081    0.11174. 1.340E-17-168.7
                           72.352 66.102                          -159.3
0.5962E+01
0.1000E+01
0.0000E+00
0.2260E-01
-.2864E-22
0.1631E-05
0.5000E+00
 DPR   -600.  -547.  -561.217057.217058.1.081    0.11174. 1.005E-15-150.0
                           72.352 66.102                          -140.5
```

For Case 1, Output B, the format is exactly as in Output A

CASE 1, NO SURFACE DUCT

INPUT USED IN EXAMPLE

ENTER THE APPROPRIATE PARAMETERS IN
THE DIMENSIONS INDICATED.

VELOCITY VECTOR X-DIRECTION(KTS)= 220.0

VELOCITY VECTOR Y-DIRECTION(KTS)= 0.0

INTIAL X-COORDINATE OF AIRCRAFT(FT)= 0.0

INITIAL Y-COORDINATE OF AIRCRAFT(FT)= 24000.0

HEIGHT OF AIRCRAFT FROM SEA SQRFACE(FT)= 10000.0

FREQ. OF AIRCRAFT RADIATION(HZ)= 150.0

VELOCITY VECTOR X-DIRECTION SUB(KTS)= 7.0

INITIAL X-COORDINATE OF SUB(FT)= 0.0

DEPTH OF SUB(FT)= 400.0

IS THE AIR TEMP. LESS THAN 50F? NO

WIND SPEED(KTS)= 8.8

X-DIRECTION COSINE= .707

Y-DIRECTION COSINE= .707

BOTTOM TYPE EITHER 3.0 OR 5.0= 3.0

INITIAL TIME= -1000.0

FINAL TIME= 1000.0

TIME INCREMENTS= 100.0

IS THERE A SURFACE DUCT NO

SURFACE SOUND SPEED(FT/SEC)= 5052.0

BOTTOM SOUND SPEED(FT/SEC)= 5053.0

DEPTH FOR BOTTOM SOUND SPEED(FT)= 15660.0

MIN. SOUND SPEED(FT/SEC)= 4875.0

DEPTH AT MIN. SOUND SPEED(FT)= 3440.0

CASE 2 OUTPUT (A): SURFACE DUCT PRESENT, DOUBLE BOUNCE OUTPUT

```
TRYING A SURFACE DUCT WITH DOUBLE BOUNCE SOLUTION
0.114E+02 0.116E+00
THE0= 0.144E+02
MODE     T     TA   TDIF      R'      R  DOP    D/E  PHI  P RATIO     TL
 BDS   640.  688.  676.230421.230421.0.931  14.41  3.  4.110E-13-123.9
                    76.807 82.519                            -116.2
0.5962E+01
0.1000E+01
0.0000E+00
0.2260E-01
-.2864E-22
0.1631E-05
0.5000E+00
0.5418E-01
0.1145E+02 0.1157E+00
0.1186E+06
 BDSR  640.  688.  676.230421.230421.0.931  14.41  3.  3.595E-13-124.4
                    76.807 82.519                            -116.8
 DP    640.  695.  684.230421.230421.0.935   0.10  3.  1.053E-17-169.8
                    76.807 83.411                            -162.1
0.5962E+01
0.1000E+01
0.0000E+00
0.2260E-01
-.2864E-22
0.1631E-05
0.5000E+00
 DPR   640.  695.  684.230421.230421.0.935   0.10  3.  0.000E+00-999.0
                    76.807 83.411                            -991.3
```

For each particular mode of propagation, two main lines of output are printed. Numbers 1-14 are the main output values. They are:

(1) Mode of propagation

(2) Running time parameter t

(3) Time of arrival (secs) t_A

(4) Difference in time between arrival time of mode t_A and time of arrival of direct path from CPA (secs)

(5) Lateral range between source and receiver (ft) at time signal leaves source (t)

(6) Slant range between source and receiver (ft) at time signal leaves source (t)

(7) Doppler shift at arrival time t_A

(8) D/E arrival angle at receiver (deg) at t_A

(9) Azimuthal arrival angle (deg) at t_A

(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses) at t_A

(11) Transmission loss re 1 ft (dB) at t_A

(12) Lateral range (kiloyards), at time signal leaves source (t)

(13) Lateral range at time of arrival (kiloyards) at t_A

(14) Transmission loss re 1 yd (dB), includes air attenuation, at t_A

Also outputted are some supporting data used as checks, they are; (15-28):

(15) Grazing angle at bottom (deg)

(16) Bottom loss from LRAPP for this grazing angle

(17) D/E angle at receiver

(18) $\xi = (8h\lambda_{AIR} g^2/\pi^2 U^4)^{1/4}$

(19) ERF(ξ)

(20) σ^2 (mean square sea slope (fully arisen sea))

(21) Most probable slope estimate (radians)

(22) $\overline{\eta^2} = 2.3\times10^4 \frac{\sqrt{2}}{gU^3} \frac{4}{3} \left[\sqrt{\pi}\{1-erf(\xi)\} + \frac{e^{-\xi^2}}{\xi}\left\{\frac{1}{2\xi^2}-1\right\}\right]$

(23) $m = h\sqrt{\overline{\eta^2}}$

(24) First part of $\overline{N}$(average number of refracting paths)

$$= \frac{m^2}{4\pi}\left[e^{-1/m^2} + \frac{\sqrt{\pi}}{2m}\left\{1+\operatorname{erf}\left(\frac{1}{m}\right)\right\}\right]^2 2$$

(25) $$\overline{N}\left\{\frac{1}{2}\left\{1+\operatorname{erf}\left(\frac{\tan\theta_0+\overline{\Psi}\cos\phi_w}{\sqrt{2}\,\sigma_w}\right)\right\}\cos^2\theta_0\left[\sigma_w^2+(\tan\theta_0+\overline{\Psi}\cos\phi_w)^2\right]+\frac{\sigma_w}{\sqrt{2\pi}}\cos^2\theta_0(\tan\theta_0+\overline{\Psi}\cos\phi_w)\cdot \exp\left[-(\tan\theta+\overline{\Psi}\cos\phi_w)^2/2\sigma_w^2\right]\right\}$$

(26) Grazing angle to bottom

(27) Bottom loss from LRAPP

(28) Path length

CASE 2, OUTPUT (B): SURFACE DUCT WITH SINGLE BOUNCE OUTPUT

```
SURFACE DUCT WITH SINGLE BOUNCE SOLUTION
 0.2173E+06-.1513E+08
       0.034     0.019      0.014
 1100.000 4990.400 4998.500 4886.500 5050.000
 0.422E+00 0.000E+00
 THEO=  0.882E+01
MODE     T     TA   TDIF      R'        R   DOP    D/E   PHI    P RATIO     TL
 BSS    620.   664.   653.223240.223240.0.929    8.82   3.  1.713E-14-137.7
                      74.413 79.695                            -130.0

 0.5962E+01
 0.1000E+01
 0.0000E+00
 0.2260E-01
 -.2864E-22
 0.1631E-05
 0.5000E+00
 0.1865E-01
 0.4216E+00 0.0000E+00
 0.2201E+06
 BSSR   620.   664.   653.223240.223240.0.929    8.82   3.  1.358E-14-138.7
                      74.413 79.695                            -131.0
 DP     620.   674.   662.223240.223240.0.935    0.10   3.  1.197E-17-169.2
                      74.413 80.845                            -161.6

 0.5962E+01
 0.1000E+01
 0.0000E+00
 0.2260E-01
 -.2864E-22
 0.1631E-05
 0.5000E+00
 DPR    620.   674.   662.223240.223240.0.935    0.10   3.  0.000E+00-999.0
                      74.413 80.845                            -991.3
```

For each particular mode of propagation, two main lines of output are printed. Numbers 1-14 are the main output values. They are:

(1) Mode of propagation

(2) Running time parameter t

(3) Time of arrival (secs) t_A

(4) Difference in time between arrival time of mode t_A and time of arrival of direct path from CPA (secs)

(5) Lateral range between source and receiver (ft) at time signal leaves source (t)

(6) Slant range between source and receiver (ft) at time signal leaves source (t)

(7) Doppler shift at arrival time t_A

(8) D/E arrival angle at receiver (deg) at t_A

(9) Azimuthal arrival angle (deg) at t_A

(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses)at t_A

(11) Transmission loss re 1 ft (dB) at t_A

(12) Lateral range (kiloyards), at time signal leaves source (t)

(13) Lateral range at time of arrival (kiloyards) at t_A

(14) Transmission loss re 1 yd (dB), includes air attenuation, at t_A

Also outputted are some supporting data used as checks, they are; (15-38):

(15) $B(\theta_0)$ - bounce distance

(16) $\partial B/\partial\theta_0$

(17) G1

(18) GS

(19) GB

(20) C1 - air sound speed

(21) CS - surface sound speed

(22) CMAX - max. sound speed

(23) CMIN - min. sound speed

(24) CB - bottom sound speed

(25) Grazing angle to bottom

(26) Bottom loss from LRAPP

(27) D/E angle at receiver

(28) ξ

(29) erf (ξ)

(30) Mean square sea slope (fully arisen sea)

(31) Most probable slope estimate (radians)

(32) $\overline{\eta^2}$

(33) m

(34) First part of $\overline{N}$(average number of refracting paths)

(35) $\overline{N}\left[\frac{1}{2}\left\{1+\operatorname{erf}\left(\frac{\tan\theta_0+\overline{\Psi}\cos\phi_w}{\sqrt{2}\,\sigma_w}\right)\right\}\cos^2\theta_0\left[\sigma_w^2+(\tan\theta_0+\overline{\Psi}\cos\phi_w)^2\right]+\frac{\sigma_w}{\sqrt{2\pi}}\cos^2\theta_0\left[\tan\theta_0+\overline{\Psi}\cos\phi_w\right]\cdot\right.$

$\left.\exp\left[-(\tan\theta_0+\overline{\Psi}\cos\phi_w)^2/2\sigma_w^2\right]\right]$

(36) Grazing angle to bottom

(37) Bottom loss from FACT

(38) Path length ($S(\theta_0)$)

CASE 2, WITH SURFACE DUCT

INPUT USED IN EXAMPLE

```
.LOAD WATER.REL,BO.REL,SBO.REL,BT3.REL,BT5.REL,ERFS.REL
LOADING

LOADER 11K CORE
13+3K MAX 84 WORDS FREE

EXIT

.ST


ENTER THE APPROPRIATE PARAMETERS IN
THE DIMENSIONS INDICATED.

VELOCITY VECTOR X-DIRECTION(KTS)=  220.0

VELOCITY VECTOR Y-DIRECTION(KTS)=  0.0

INTIAL X-COORDINATE OF AIRCRAFT( FT)=  0.0

INITIAL Y-COORDINATE OF AIRCRAFT(FT)=  12000.0

HEIGHT OF AIRCRAFT FROM SEA SQRFACE(FT)=  10000.0

FREQ. OF AIRCRAFT RADIATION(HZ)=  150.0

VELOCITY VECTOR X-DIRECTION SUB(KTS)=  7.0

INITIAL X-COORDINATE OF SUB(FT)=  0.0

DEPTH OF SUB(FT)=  400.0

IS THE AIR TEMP. LESS THAN 50F?  YES

WIND SPEED(KTS)=  8.8

X-DIRECTION COSINE=  .707

Y-DIRECTION COSINE=  .707

BOTTOM TYPE EITHER 3.0 OR 5.0=  3.0

INITIAL 'TIME=  600.0

FINAL TIME=  700.0
```

Case 2 continued.

```
TIME INCREMENTS=  20.0
IS THERE A SURFACE DUCT  YES
SURFACE SOUND SPEED(FT/SEC)=  4990.4
BOTTOM SOUND SPEED(FT/SEC)=  5050.0
DEPTH FOR BOTTOM SOUND SPEED(FT)=  15540.0
MIN. SOUND SPEED(FT/SEC)=  4886.5
DEPTH AT MIN. SOUND SPEED(FT)=  3720.0
MAX. SOUND SPEED(FT/SEC)=  4998.5
DEPTH AT MAX. SOUND SPEED(FT)=  420.0
```

B.5 Program Listing

```
TYPE WATER,\,\.F4
        COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT,U,WSX,WSY
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PPSQ,A,MODE
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
        1 G1
        M=0.0
        TYPE 1
1       FORMAT(1H ,'ENTER THE APPROPRIATE PARAMETERS IN'/
        1' THE DIMENSIONS INDICATED.',/)
        CALL AIRV(VAX,VAY)
        CALL AIRC(XIA,YIA)
        CALL AIRH(H)
        CALL AIRF(F)
        CALL SUBV(VS)
        CALL SUBC(XIS)
        CALL SUBD(D)
        TYPE 6666
6666    FORMAT(1H ,'IS THE AIR TEMP. LESS THAN 50F?',2X,$)
        ACCEPT 6665,LANS
6665    FORMAT(A3)
        IF(LANS.EQ.'YES') GO TO 6667
        AL=(.000000074*(F**2)*H)/(50.0*3.281)
        GO TO 6664
6667    AL=(.00125*F*H)/1000.0
6664    CONTINUE
        CALL WIND(U,WSX,WSY)
        CALL BOT(BT)
        CALL TIM(TI,TF,DT)
        TYPE 2
2       FORMAT(1H ,'IS THERE A SURFACE DUCT',2X,$)
        ACCEPT 3,LANS
3       FORMAT(A3)
        IF(LANS.EQ.'YES') GO TO 4
        CALL ENVS(CS)
        CALL ENVC(CB,DB)
        CALL ENVD(CMIN,DMIN)
        GO TO 5
4       CALL ENVS(CS)
        CALL ENVC(CB,DB)
        CALL ENVD(CMIN,DMIN)
        CALL ENVA(CMAX,DS)
5       CONTINUE
30      CONTINUE
        T=TI
911     CONTINUE
        GS=(CS-CMIN)/DMIN
        GB=(CB-CMIN)/(DB-DMIN)
        C1=1100.0
        AXN=.22
        C2=5000.0
        PI=3.14159265
        TX1=(XIA-XIS)
        TX2=(VS-VAX)
        TX3=(YIA*VAY)
        TX4=(TX1*TX2)-TX3
        TX5=TX2**2
        TX6=TX5+(VAY**2)
        TO=TX4/TX6
        RX1=(VS*TO)+XIS
        RX2=(VAX*TO)+XIA
        RX3=(RX1-RX2)**2
        RX4=((VAY*TO)+YIA)**2
        RO=SQRT(RX3+RX4+(D**2))
```

```
      THE0=0.0
      AX1=(F/1000.0)**2
      AX2=(0.1*AX1)/(1.0+AX1)
      AX3=(40.0*AX1)/(4100.0+AX1)
      A=(AX2+AX3)/3000.0
      RY1=(VS*T)+XIS
      RY2=(VAX*T)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*T)+YIA)**2
      RP=SQRT(RY3+RY4)
      R=SQRT(RY3+RY4+(D**2))
      IF(LANS.EQ.'YES') GO TO 333
      CALL BOUN(THE0)
      GO TO 334
333   G1=(CMAX-CMIN)/(DMIN-DS)
      GS=(CMAX-CS)/DS
      CALL SBOUN(THE0)
334   TYPE 31,THE0
31    FORMAT(1H ,'THE0=',E10.3)
      TYPE 703
703   FORMAT(1H ,'MODE     T     TA   TDIF      R''       R   DOP   D/E'
     1 '   PHI    P RATIO    TL')
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
705   FORMAT(1H ,1XA4,3F6.0,2F7.0,F5.3,F7.2,F4.0,1PE10.3,0PF6.1)
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
706   FORMAT(1H ,23X,2F7.3,20X,0PF6.1)
      IF(MODE.EQ.'BS') GO TO 4444
      IF(MODE.EQ.'BD') GO TO 4446
      IF(MODE.EQ.'BSS') GO TO 4447
      IF(MODE.EQ.'BDS') GO TO 4448
      GO TO 4445
4444  CALL BSR(THE0)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
      GO TO 4445
4446  CALL BDR(THE0)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
```

```
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
      GO TO 4445
4447  CALL BSSR(THEO)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
      GO TO 4445
4448  CALL BDSR(THEO)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
4445  CALL DP(THEO)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
      CALL DPR(THEO)
      TYPE 705,MODE,T,TA,TDIF,RP,R,DOP,DE,PHI,PRSQ,TL
      RPK=RP/3000.0
      RY1=(VS*TA)+XIS
      RY2=(VAX*TA)+XIA
      RY3=(RY1-RY2)**2
      RY4=((VAY*TA)+YIA)**2
      RPA=SQRT(RY3+RY4)
      RPT=RPA/3000.0
      TL1=TL+(10.0*ALOG10(9.0))-AL
      TYPE 706,RPK,RPT,TL1
      T=T+DT
      IF(T.LT.TF) GO TO 911
      M=M+1
      IF(M.GT.1) GO TO 3000
      TYPE 2000
2000  FORMAT(1H ,'TO CHANGE A RUN PARAMETER, ENTER THE'/
     1 ' APPROPRIATE NUMBER:'/
     1 '    1  AIRCRAFT VEL. VECTORS'/
     1 '    2  AIRCRAFT INITIAL POSITION'/
     1 '    3  AIRCRAFT HEIGHT'/
```

```
         1 '   4  AIRCRAFT RADIATED FREQ.'/
         1 '   5  SUB VEL. VECTOR'/
         1 '   6  SUB INITIAL POSITION'/
         1 '   7  SUB DEPTH'/
         1 '   8  SURFACE SOUND SPEED'/
         1 '   9  BOTTOM SOUND SPEED AND DEPTH'/
         1 '   10 MINIMUM SOUND SPEED AND DEPTH'/
         1 '   11 WIND SPEED AND DIRECTION COSINES'/
         1 '   12 BOTTOM TYPE'/
         1 '   13 MAX. SOUND SPEED AND DEPTH'/
         1 '   14 TIME OF EVENTS'/
         1 '   15 RUN'/
         1 '   16 STOP'/
         1 ' ENTER THE APPROPRIATE NUMBER = ',$)
         ACCEPT 2001,K3
2001     FORMAT(I)
         GO TO (47,48,49,10,11,12,13,14,15,16,50,17,19,20,30,18)K3
3000     TYPE 3001
3001     FORMAT(1H ,'ENTER THE NUMBER OF THE PARAMETER'/
         1 ' THAT YOU WISH TO CHANGE = ',$)
         ACCEPT 3002,K4
3002     FORMAT(I)
         GO TO (47,48,49,10,11,12,13,14,15,16,50,17,19,20,30,18)K4
47       CALL AIRV(VAX,VAY)
         GO TO 3000
48       CALL AIRC(XIA,YIA)
         GO TO 3000
49       CALL AIRH(H)
         GO TO 3000
10       CALL AIRF(F)
         GO TO 3000
11       CALL SUBV(VS)
         GO TO 3000
12       CALL SUBC(XIS)
         GO TO 3000
13       CALL SUBD(D)
         GO TO 3000
14       CALL ENVS(CS)
         GO TO 3000
15       CALL ENVC(CB,DB)
         GO TO 3000
16       CALL ENVD(CMIN,DMIN)
         GO TO 3000
50       CALL WIND(U,WSX,WSY)
         GO TO 3000
17       CALL BOT(BT)
         GO TO 3000
19       CALL ENVA(CMAX,DS)
         GO TO 3000
20       CALL TIM(TI,TF,DT)
         GO TO 3000
18       CALL EXIT
         END
         SUBROUTINE AIRV(VAX,VAY)
         TYPE 200
200      FORMAT(1H ,'VELOCITY VECTOR X-DIRECTION(KTS)=',2X,$)
         ACCEPT 201,VAXKT
201      FORMAT(F)
         VAX=1.688*VAXKT
         TYPE 202
202      FORMAT(1H ,'VELOCITY VECTOR Y-DIPECTION(KTS)=',2X,$)
```

```
      ACCEPT 201,VAYKT
      VAY=1.688*VAYKT
      RETURN
      END
      SUBROUTINE AIRC(XIA,YIA)
      TYPE 200
200   FORMAT(1H ,'INTIAL X-COORDINATE OF AIRCRAFT( FT)=',2X,$)
      ACCEPT 201,XIA
201   FORMAT(F)
      TYPE 202
202   FORMAT(1H ,'INITIAL Y-COORDINATE OF AIRCRAFT(FT)=',2X,$)
      ACCEPT 201,YIA
      RETURN
      END
      SUBROUTINE AIRH(H)
      TYPE 200
200   FORMAT(1H ,'HEIGHT OF AIRCRAFT FROM SEA SORFACE(FT)=',2X,$)
      ACCEPT 201,H
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE AIRF(F)
      TYPE 200
200   FORMAT(1H ,'FREQ. OF AIRCRAFT RADIATION(HZ)=',2X,$)
      ACCEPT 201,F
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE SUBV(VS)
      TYPE 200
200   FORMAT(1H ,'VELOCITY VECTOR X-DIRECTION SUB(KTS)=',2X,$)
      ACCEPT 201,VSKT
201   FORMAT(F)
      VS=1.688*VSKT
      RETURN
      END
      SUBROUTINE SUBC(XIS)
      TYPE 200
200   FORMAT(1H ,'INITIAL X-COORDINATE OF SUB(FT)=',2X,$)
      ACCEPT 201,XIS
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE SUBD(D)
      TYPE 200
200   FORMAT(1H ,'DEPTH OF SUB(FT)=',2X,$)
      ACCEPT 201,D
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE ENVS(CS)
      TYPE 200
200   FORMAT(1H ,'SURFACE SOUND SPEED(FT/SEC)=',2X,$)
      ACCEPT 201,CS
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE ENVC(CB,DB)
      TYPE 200
200   FORMAT(1H ,'BOTTOM SOUND SPEED(FT/SEC)=',2X,$)
```

```
      ACCEPT 201,CB
201   FORMAT(F)
      TYPE 202
202   FORMAT(1H ,'DEPTH FOR BOTTOM SOUND SPEED(FT)=',2X,$)
      ACCEPT 201,DB
      RETURN
      END
      SUBROUTINE ENVD(CMIN,DMIN)
      TYPE 200
200   FORMAT(1H ,'MIN. SOUND SPEED(FT/SEC)=',2X,$)
      ACCEPT 201,CMIN
201   FORMAT(F)
      TYPE 202
202   FORMAT(1H ,'DEPTH AT MIN. SOUND SPEED(FT)=',2X,$)
      ACCEPT 201,DMIN
      RETURN
      END
      SUBROUTINE ENVA(CMAX,DS)
      TYPE 200
200   FORMAT(1H ,'MAX. SOUND SPEED(FT/SEC)=',2X,$)
      ACCEPT 201,CMAX
201   FORMAT(F)
      TYPE 202
202   FORMAT(1H ,'DEPTH AT MAX. SOUND SPEED(FT)=',2X,$)
      ACCEPT 201,DS
      RETURN
      END
      SUBROUTINE TIM(TI,TF,DT)
      TYPE 200
200   FORMAT(1H ,'INITIAL TIME=',2X,$)
      ACCEPT 201,TI
201   FORMAT(F)
      TYPE 202
202   FORMAT(1H ,'FINAL TIME=',2X,$)
      ACCEPT 201,TF
      TYPE 203
203   FORMAT(1H ,'TIME INCREMENTS=',2X,$)
      ACCEPT 201,DT
      RETURN
      END
      SUBROUTINE BOT(BT)
      TYPE 200
200   FORMAT(1H ,'BOTTOM TYPE EITHER 3.0 OR 5.0=',2X,$)
      ACCEPT 201,BT
201   FORMAT(F)
      RETURN
      END
      SUBROUTINE WIND(U,WSX,WSY)
      TYPE 200
200   FORMAT(1H ,'WIND SPEED(KTS)=',2X,$)
      ACCEPT 201,U1
201   FORMAT(F)
      U=U1*1.688
      TYPE 202
202   FORMAT(1H ,'X-DIRECTION COSINE=',2X,$)
      ACCEPT 201,WSX
      TYPE 203
203   FORMAT(1H ,'Y-DIRECTION COSINE=',2X,$)
      ACCEPT 201,WSY
      RETURN
      END
```

```
      SUBROUTINE DP(THEO)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT,U,WSX,WSY
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      X1=(VS*T)+XIS
      X2=(VAX*T)+XIA
      X3=(X1-X2)**2
      X4=(VAY*T)+YIA
      X5=X4**2
      SIN=D/SQRT(X3+X5+(D**2))
      DE1=ASIN(SIN)
      DE=(DE1*180.0)/PI
      MODE='DP'
      CO1=(VAX*T)+XIA
      CO2=X1
      CO3=(CO1-X1)/SQRT(X3+X5)
      CO4=ACOS(CO3)
      PHI=(CO4*180.0)/PI
      TDIF=T-TO+(R-RO)/C2
      TA=T+(H/C1)+(R/C2)
      ZX=(VS*T)+XIS
      ZX1=(VAX*T)+XIA
      ZX2=(VAY*T)+YIA
      ZX3=(VAX*(ZX-ZX1))-(VAY*ZX2)
      ZX4=SQRT(VAX**2+VAY**2)
      ZX5=ZX3/(ZX4*R)
      DO1=(2.0*ZX4*ZX5)/C2
      DO2=SQRT(1.0-DO1)
      DO3=1.0/DO2
      DO5=(ZX-ZX1)/R
      DO6=(2.0*VS*DO5)/C2
      DO7=SQRT(1.0-DO6)
      DOP=(DO3*DO7)
      QZ1=-(A*R)/10.0
      QZ2=(10.0**QZ1)
      PR1=4.0*(AXN**2)*(D**2)*QZ2
      PRSQ=PR1/(R**4)
      ETA=1.0/(10.0**20)
      IF(PRSQ.LT.ETA) GO TO 999
      GO TO 998
999   TL=-999.0
      GO TO 100
998   TL=(10.0*ALOG10(PRSQ))
100   RETURN
      END
      SUBROUTINE DPR(THEO)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT,U,WSX,WSY
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      MODE='DPR'
      COS1=(VS*T)+XIS
      COS2=(VAX*T)+XIA
      COS3=(VAY*T)+YIA
      COS4=((WSX*(COS1-COS2))-(WSY*COS3))/RP
      ZET=C1/F
      ZET1=(8.0*H*ZET*(32.174**2))/((PI**2)*(U**4))
      ZET2=SQRT(ZET1)
      ZET3=SQRT(ZET2)
```

```
      TYPE 1111,ZET3
1111  FORMAT(1H ,E10.4)
      ZET4=U*30.48
      CALL ERFS(ZET3,ERF)
      TYPE 1111,ERF
      SIG=(11500.0*ZET4)/(981.5**3)
      SIG1=SQRT(PI/2.0)
      SIG2=(SIG*SIG1)*(1.0-ERF)
      TYPE 1111,SIG2
      SIGW=SQRT(SIG2)/SQRT(2.0)
      CY=(2.86*ZET4)/1000.0
      CY1=(CY*PI)/180.0
      TYPE 1111,CY1
      AT1=(23000.0*(SQRT(2.0))*4.0)/(981.5*(ZET4**3)*3.0)
      AT2=(SQRT(PI))*(1.0-ERF)
      AT3=ZET3**2
      AT4=EXP(-AT3)/ZET3
      AT5=(1.0/(2.0*AT3))-1.0
      ATN=AT1*(AT2+(AT4*AT5))
      TYPE 1111,ATN
      HX=H*30.48
      EM=HX*SQRT(ATN)
      TYPE 1111,EM
      EM1=1.0/EM
      CALL ERFS(EM1,ERF)
      BN1=((1.0+ERF)*SQRT(PI))/(2.0*EM)
      BN2=(1.0/(EM**2))
      BN3=((EXP(-BN2)+BN1)**2)*2.0
      BN4=((EM**2)+BN3)/(4.0*PI)
      TYPE 1111,BN4
      SXN=D/R
      SXN1=ASIN(SXN)
      SXN2=(180.0*SXN1)/PI
      SXN3=SIND(SXN2)/COSD(SXN2)
      SXN4=SXN3/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN4,ERF)
      SXN5=ERF
      SXN6=(1.0-SIND(SXN2))/COSD(SXN2)
      SXN7=SXN6/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN7,ERF)
      BN5=SXN5+ERF
      BN6=BN5*BN4
      FO1=SXN3+(CY1*COS4)
      FO2=FO1/((SQRT(2.0))*SIGW)
      CALL ERFS(FO2,ERF)
      FO3=(1.0+ERF)/2.0
      FO4=((SIGW**2)+(FO1**2))*(COSD(SXN2))**2
      FO5=(FO4*FO3)
      FO6=((FO1)**2)/(2.0*(SIGW**2))
      FO7=EXP(-FO6)
      FO8=(((COSD(SXN2))**2)*FO7*SIGW)/SQRT(2.0*PI)
      FO9=(FO5+FO8)*BN6
      FO10=-(A*R)/10.0
      FO11=(10.0**FO10)
      FO12=FO9*FO11
      PO1=(2.0*AXN)/R
      PRSQ=(PO1**2)*FO12
      ETA=1.0/(10.0**20)
      IF(PRSQ.LT.ETA) GO TO 999
```

```
      GO TO 998
999   TL=-999.0
      GO TO 100
998   TL=10.0*ALOG10(PRSQ)
100   RETURN
      END
      SUBROUTINE BSR(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXM,BT,U,WSX,WSY
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
      1 G1
      B1=((CMIN/CS)**2)*(COSD(THE0))**2
      B2=(SQRT(1.0-B1))-SIND(THE0)
      B3=((CB/CS)**2)*(COSD(THE0))**2
      B4=SQRT(1.0-B3)
      B5=(SQRT(1.0-B1))-B4
      B6=(B2/GS)+(B5/GB)
      B7=(2.0*CS*B6)/COSD(THE0)
      T1=CS/(CMIN*COSD(THE0))
      T2=ALOG(T1+SQRT((T1**2)-1.0))
      T3=1.0/COSD(THE0)
      T4=ALOG(T3+SQRT((T3**2)-1.0))
      T5=((T2-T4)*2.0)/GS
      T6=CS/(CB*COSD(THE0))
      T7=ALOG(T6+SQRT((T6**2)-1.0))
      T8=((T7-T2)*2.0)/GB
      T9=T5-T8
      MODE='BSR'
      P1=B7*(SIND(THE0)/COSD(THE0))
      P2=((CMIN/CS)**2)*SIND(THE0)
      P3=((CMIN/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=((P2/P4)-1.0)/GS
      P6=(CB/CS)**2
      P7=P6*(COSD(THE0))**2
      P8=SQRT(1.0-P7)
      P9=P6/P8
      P10=(((CMIN/CS)**2)/P4)-P9
      P11=((SIND(THE0))*P10)/GB
      P12=((P5+P11)*2.0*CS)+P1
      COS1=(VS*(T+T9))+XIS
      COS2=(VAX*(T-(H/C1)))+XIA
      COS3=(VAY*(T-(H/C1)))+YIA
      COS4=((WSX*(COS1-COS2))-(WSY*COS3))/B7
      ZET=C1/F
      ZET1=(8.0*H*ZET*(32.174**2))/((PI**2)*(U**4))
      ZET2=SQRT(ZET1)
      ZET3=SQRT(ZET2)
      TYPE 1111,ZET3
1111  FORMAT(1H ,E10.4)
      ZET4=U*30.48
      CALL ERFS(ZET3,ERF)
      TYPE 1111,ERF
      SIG=(11500.0*ZET4)/(981.5**3)
      SIG1=SQRT(PI/2.0)
      SIG2=(SIG*SIG1)*(1.0-ERF)
      TYPE 1111,SIG2
      SIGW=SQRT(SIG2)/SQRT(2.0)
      CY=(2.86*ZET4)/1000.0
      CY1=(CY*PI)/180.0
      TYPE 1111,CY1
```

```
      AT1=(23000.0*(SQRT(2.0))*4.0)/(981.5*(ZET4**3)*3.0)
      AT2=(SQRT(PI))*(1.0-ERF)
      AT3=ZET3**2
      AT4=EXP(-AT3)/ZET3
      AT5=(1.0/(2.0*AT3))-1.0
      ATN=AT1*(AT2+(AT4*AT5))
      TYPE 1111,ATN
      HX=(H*30.48)
      EM=HX*SQRT(ATN)
      TYPE 1111,EM
      EM1=(1.0/EM)
      CALL ERFS(EM1,ERF)
      BN1=((1.0+ERF)*SQRT(PI))/(2.0*EM)
      BN2=(1.0/(EM**2))
      BN3=((EXP(-BN2)+BN1)**2)*2.0
      BN4=((EM**2)*BN3)/(4.0*PI)
      TYPE 1111,BN4
      SXN3=SIND(THE0)/COSD(THE0)
      SXN4=SXN3/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN4,ERF)
      SXN5=ERF
      SXN6=(1.0-SIND(THE0))/COSD(THE0)
      SXN7=SXN6/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN7,ERF)
      BN5=SXN5+ERF
      BN6=BN5*BN4
      FO1=SXN3+(CY1*COS4)
      FO2=FO1/((SQRT(2.0))*SIGW)
      CALL ERFS(FO2,ERF)
      FO3=(1.0-ERF)/2.0
      FO4=((SIGW**2)+(FO1**2))*(COSD(THE0))**2
      FO5=(FO4*FO3)
      FO6=((FO1)**2)/(2.0*(SIGW**2))
      FO7=EXP(-FO6)
      FO8=(((COSD(THE0))**2)*FO7*SIGW)/SQRT(2.0*PI)
      FO9=(FO5+FO8)*BN6
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,RBL)
      GO TO 9997
9999  CALL BT5(TB2,RBL)
9997  TYPE 9995,TB2,RBL
9995  FORMAT(1H ,2E10.4)
      S1=(2.0*CS)/COSD(THE0)
      S2=(1.0/GS)+(1.0/GB)
      S3=ASIN((CMIN*COSD(THE0))/CS)
      S4=ASIN((CB*COSD(THE0))/CS)
      S5=((PI/2.0)-((THE0*PI)/180.0))/GS
      S6=((-S2*S3)+(S4/GB)+S5)*S1
      FO10=-(RBL+(A*S6))/10.0
      FO11=(10.0**FO10)
      FO12=FO9*FO11
      PQ1=(8.0*(AXN**2)*COSD(THE0))/((SIND(THE0))*B7*ABS(P12))
      PRSQ=PQ1*FO12
      TL=10.0*ALOG10(PRSQ)
      RETURN
      END
```

```
      SUBROUTINE BDR(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT,U,WSX,WSY
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CL,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RD,TD,PI,
     1 G1
      B1=((CMIN/CS)**2)*(COSD(THE0))**2
      B2=(SQRT(1.0-B1))-SIND(THE0)
      B3=((CB/CS)**2)*(COSD(THE0))**2
      B4=SQRT(1.0-B3)
      B5=(SQRT(1.0-B1))-B4
      B6=(B2/GS)+(B5/GB)
      B7=(2.0*CS*B6)/COSD(THE0)
      T1=CS/(CMIN*COSD(THE0))
      T2=ALOG(T1+SQRT((T1**2)-1.0))
      T3=1.0/COSD(THE0)
      T4=ALOG(T3+SQRT((T3**2)-1.0))
      T5=((T2-T4)*2.0)/GS
      T6=CS/(CB*COSD(THE0))
      T7=ALOG(T6+SQRT((T6**2)-1.0))
      T8=((T7-T2)*2.0)/GB
      T9=T5-T8
      MODE='BDR'
      P1=B7*(SIND(THE0)/COSD(THE0))
      P2=((CMIN/CS)**2)*SIND(THE0)
      P3=((CMIN/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=((P2/P4)-1.0)/GS
      P6=(CB/CS)**2
      P7=P6*(COSD(THE0))**2
      P8=SQRT(1.0-P7)
      P9=P6/P8
      P10=(((CMIN/CS)**2)/P4)-P9
      P11=((SIND(THE0))*P10)/GB
      P12=((P5+P11)*2.0*CS)+P1
      COS1=(VS*(T+(2.0*T9)))+XIS
      COS2=(VAX*(T-(H/C1)))+XIA
      COS3=(VAY*(T-(H/C1)))+YIA
      COS4=((WSX*(COS1-COS2))-(WSY*COS3))/(2.0*B7)
      ZET=C1/F
      ZET1=(8.0*H*ZET*(32.174**2))/((PI**2)*(U**4))
      ZET2=SQRT(ZET1)
      ZET3=SQRT(ZET2)
      TYPE 1111,ZET3
 1111 FORMAT(1H ,E10.4)
      ZET4=U*30.48
      CALL ERFS(ZET3,ERF)
      TYPE 1111,ERF
      SIG=(11500.0*ZET4)/(981.5**3)
      SIG1=SQRT(PI/2.0)
      SIG2=(SIG*SIG1)*(1.0-ERF)
      TYPE 1111,SIG2
      SIGW=SQRT(SIG2)/SQRT(2.0)
      CY=(2.86*ZET4)/1400.0
      CY1=(CY*PI)/180.0
      TYPE 1111,CY1
      AT1=(23000.0*(SQRT(2.0))*4.0)/(981.5*(ZET4**3)*3.0)
      AT2=(SQRT(PI))*(1.0-ERF)
      AT3=ZET3**2
      AT4=EXP(-AT3)/ZET3
      AT5=(1.0/(2.0*AT3))-1.0
      ATN=AT1*(AT2+(AT4*AT5))
```

```
      TYPE 1111,ATN
      HX=H*30.48
      EM=HX*SQRT(ATN)
      TYPE 1111,EM
      EM1=(1.0/EM)
      CALL ERFS(EM1,ERF)
      BN1=((1.0+ERF)*SQRT(PI))/(2.0*EM)
      BN2=(1.0/(EM**2))
      BN3=((EXP(-BN2)+BN1)**2)*2.0
      BN4=((EM**2)*BN3)/(4.0*PI)
      TYPE 1111,BN4
      SXN3=SIND(THE0)/COSD(THE0)
      SXN4=SXN3/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN4,ERF)
      SXN5=ERF
      SXN6=(1.0-SIND(THE0))/COSD(THE0)
      SXN7=SXN6/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN7,ERF)
      BN5=(SXN5+ERF)
      BN6=BN5*BN4
      FO1=SXN3+(CY1*COS4)
      FO2=FO1/((SQRT(2.0))*SIGW)
      CALL ERFS(FO2,ERF)
      FO3=(1.0+ERF)/2.0
      FO4=((SIGW**2)+(FO1**2))*(COSD(THE0))**2
      FO5=(FO4*FO3)
      FO6=((FO1)**2)/(2.0*(SIGW**2))
      FO7=EXP(-FO6)
      FO8=(((COSD(THE0))**2)*FO7*SIGW)/SQRT(2.0*PI)
      FO9=(FO5+FO8)*BN6
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,PBL) .
      GO TO 9997
9999  CALL BT5(TB2,PBL)
9997  TYPE 9995,TB2,PBL
9995  FORMAT(1H ,2E10.4)
      S1=(2.0*CS)/COSD(THE0)
      S2=(1.0/GS)+(1.0/GB)
      S3=ASIN((CMIN*COSD(THE0))/CS)
      S4=ASIN((CB*COSD(THE0))/CS)
      S5=((PI/2.0)-((THE0*PI)/180.0))/GS
      S6=((-S2*S3)+(S4/GB)+S5)*S1
      FO10=-((2.0*PBL)+(2.0*A*S6))/10.0
      FO11=(10.0**FO10)
      FO12=FO9*FO11
      PO1=(8.0*(A/N**2)*COSD(THE0))/(4.0*SIND(THE0)*BT*ABS(P12))
      PRSQ=PO1*FO12
      TL=10.0*ALOG10(PRSQ)
      RETURN
      END
      SUBROUTINE BSIR(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VI,[illegible],D,ACN,BT,U,WIX,WIY
      COMMON /OUT/ T,P,PP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,PO,TO,PI,
     1 G1
      B1=(2.0*CS)/COSD(THE0)
      B2=((CMAX/CS)**2)*(COSD(THE0))**2
      B3=SQRT(1.0-B2)
```

```
      B4=(SIND(THE0)-B3)/GS
      B5=((CMIN/CS)**2)*(COSD(THE0))**2
      B6=SQRT(1.0-B5)
      B7=(B6-B3)/G1
      B8=((CB/CS)**2)*(COSD(THE0))**2
      B9=SQRT(1.0-B8)
      B10=(B6-B9)/GB
      B11=(B4+B7+B10)*B1
      TH=ALOG((1.0/COSD(THE0))+SQRT(((1.0/COSD(THE0))**2)-1.0))
      TH1=CS/(CMAX*COSD(THE0))
      TH2=ALOG(TH1+SQRT((TH1**2)-1.0))
      TH3=((TH-TH2)*2.0)/GS
      TH4=CS/(CMIN*COSD(THE0))
      TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
      TH6=((TH5-TH2)*2.0)/G1
      TH7=CS/(CB*COSD(THE0))
      TH8=ALOG(TH7+SQRT((TH7**2)-1.0))
      TH9=((TH5-TH8)*2.0)/GB
      TH10=TH3+TH6+TH9
      MODE='BSSR'
      P1=B11*(SIND(THE0)/COSD(THE0))
      P2=((CMAX/CS)**2)*(SIND(THE0))
      P3=((CMAX/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=(1.0-(P2/P4))/GS
      P6=(CMIN/CS)**2
      P7=SQRT(1.0-(P6*((COSD(THE0))**2)))
      P8=P6/P7
      P9=(CMAX/CS)**2
      P10=P9/P4
      P11=((P8-P10)*SIND(THE0))/G1
      P12=(CB/CS)**2
      P13=SQRT(1.0-(P12*((COSD(THE0))**2)))
      P14=P12/P13
      P16=((P8-P14)*SIND(THE0))/GB
      P15=((P5+P11+P16)*2.0*CS)+P1
      COS1=(VS*(T+TH10))+XIS
      COS2=(VAX*(T-(H/C1)))+XIA
      COS3=(VAY*(T-(H/C1)))+YIA
      COS4=((WIX*(COS1-COS2))-(WIY*COS3))/B11
      ZET=C1/F
      ZET1=(8.0*H*ZET*(32.174**2))/((PI**2)*(U**4))
      ZET2=SQRT(ZET1)
      ZET3=SQRT(ZET2)
      TYPE 1111,ZET3
1111  FORMAT(1H ,E10.4)
      ZET4=U*30.48
      CALL ERFS(ZET3,ERF)
      TYPE 1111,ERF
      SIG=(11500.0*ZET4)/(981.5**3)
      SIG1=SQRT(PI/2.0)
      SIG2=(SIG*SIG1)*(1.0-ERF)
      TYPE 1111,SIG2
      SIGW=SQRT(SIG2)/SQRT(2.0)
      CY=(2.86*ZET4)/1000.0
      CY1=(CY*PI)/180.0
      TYPE 1111,CY1
      AT1=(23000.0*(SQRT(2.0))*4.0)/(981.5*(ZET4**3)*3.0)
      AT2=(SQRT(PI))*(1.0-ERF)
      AT3=ZET3**2
      AT4=EXP(-AT3)/ZET3
```

```
      AT5=(1.0/(2.0*AT3))-1.0
      ATN=AT1*(AT2+(AT4*AT5))
      TYPE 1111,ATN
      HX=H*30.48
      EM=HX*SQRT(ATN)
      TYPE 1111,EM
      EM1=(1.0/EM)
      CALL ERFS(EM1,ERF)
      BN1=((1.0+ERF)*SQRT(PI))/(2.0*EM)
      BN2=(1.0/(EM**2))
      BN3=((EXP(-BN2)+BN1)**2)*2.0
      BN4=((EM**2)*BN3)/(4.0*PI)
      TYPE 1111,BN4
      SXN3=SIND(THE0)/COSD(THE0)
      SXN4=SXN3/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN4,ERF)
      SXN5=ERF
      SXN6=(1.0-SIND(THE0))/COSD(THE0)
      SXN7=SXN6/((SQRT(2.0))*SIGW)
      CALL ERFS(SXN7,ERF)
      BN5=SXN5+ERF
      BN6=BN5*BN4
      FO1=SXN3+(CY1*COS4)
      FO2=FO1/((SQRT(2.0))*SIGW)
      CALL ERFS(FO2,ERF)
      FO3=(1.0+ERF)/2.0
      FO4=((SIGW**2)+(FO1**2))*(COSD(THE0))**2
      FO5=(FO4*FO3)
      FO6=((FO1)**2)/(2.0*(SIGW**2))
      FO7=EXP(-FO6)
      FO8=(((COSD(THE0))**2)*FO7*SIGW)/SQRT(2.0*PI)
      FO9=(FO5+FO8)*BN6
      TYPE 3344,FO9
3344  FORMAT(1H ,E10.4)
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,RBL)
      GO TO 9997
9999  CALL BT5(TB2,RBL)
9997  TYPE 9995,TB2,RBL
9995  FORMAT(1H ,2E10.4)
      S1=(2.0*CS)/COSD(THE0)
      S2=ACOS((CMAX*COSD(THE0))/CS)
      S3=(((THE0*PI)/180.0)-S2)/G3
      S4=ACOS((CMIN*COSD(THE0))/CS)
      S5=(S4-S2)/G1
      S6=ACOS((CB*COSD(THE0))/CS)
      S7=(S4-S6)/GB
      S8=(S3+S5+S7)*S1
      TYPE 6543,S8
6543  FORMAT(1H ,E10.4)
      FO10=-(RBL+(A*S8))/10.0
      FO11=(10.0**FO10)
      FO12=FO9*FO11
      PO1=(8.0*(AMN**2)*COSD(THE0))/((SIND(THE0))*B11*ABS(P15))
      PRSO=PO1*FO12
      TL=10.0*ALOG10(PRSO)
      RETURN
      END
```

```
      SUBROUTINE BDSR(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT,U,WSX,WSY
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      B1=(2.0*CS)/COSD(THE0)
      B2=((CMAX/CS)**2)*(COSD(THE0))**2
      B3=SQRT(1.0-B2)
      B4=(SIND(THE0)-B3)/GS
      B5=((CMIN/CS)**2)*(COSD(THE0))**2
      B6=SQRT(1.0-B5)
      B7=(B6-B3)/G1
      B8=((CB/CS)**2)*(COSD(THE0))**2
      B9=SQRT(1.0-B8)
      B10=(B6-B9)/GB
      B11=(B4+B7+B10)*B1
      TH=ALOG((1.0/COSD(THE0))+SQRT(((1.0/COSD(THE0))**2)-1.0))
      TH1=CS/(CMAX*COSD(THE0))
      TH2=ALOG(TH1+SQRT((TH1**2)-1.0))
      TH3=((TH-TH2)*2.0)/GS
      TH4=CS/(CMIN*COSD(THE0))
      TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
      TH6=((TH5-TH2)*2.0)/G1
      TH7=CS/(CB*COSD(THE0))
      TH8=ALOG(TH7+SQRT((TH7**2)-1.0))
      TH9=((TH5-TH8)*2.0)/GB
      TH10=TH3+TH6+TH9
      MODE='BDSR'
      P1=B11*(SIND(THE0)/COSD(THE0))
      P2=((CMAX/CS)**2)*SIND(THE0)
      P3=((CMAX/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=(1.0-(P2/P4))/GS
      P6=(CMIN/CS)**2
      P7=SQRT(1.0-(P6*((COSD(THE0))**2)))
      P8=P6/P7
      P9=(CMAX/CS)**2
      P10=P9/P4
      P11=((P8-P10)*SIND(THE0))/G1
      P12=(CB/CS)**2
      P13=SQRT(1.0-(P12*((COSD(THE0))**2)))
      P14=P12/P13
      P16=((P8-P14)*SIND(THE0))/GB
      P15=((P5+P11+P16)*2.0*CS)+P1
      COS1=(VS*(T+(2.0*TH10)))+XIS
      COS2=(VAX*(T-(H/C1)))+XIA
      COS3=(VAY*(T-(H/C1)))+YIA
      COS4=((WSX*(COS1-COS2))-(WSY*COS3))/(2.0*B11)
      ZET=C1/F
      ZET1=(8.0*H*ZET*(32.174**2))/((PI**2)*(U**4))
      ZET2=SQRT(ZET1)
      ZET3=SQRT(ZET2)
      TYPE 1111,ZET3
 1111 FORMAT(1H ,E10.4)
      ZET4=U*30.48
      CALL ERFS(ZET3,ERF)
      TYPE 1111,ERF
      SIG=(11500.0*ZET4)/(981.5**3)
      SIG1=SQRT(PI/2.0)
      SIG2=(SIG*SIG1)*(1.0-ERF)
      TYPE 1111,SIG2
      SIGW=SQRT(SIG2)/SQRT(2.0)
```

```
        CY=(2.86*ZET4)/1000.0
        CY1=(CY*PI)/180.0
        TYPE 1111,CY1
        AT1=(23000.0*(SQRT(2.0))*4.0)/(981.5*(ZET4**3)*3.0)
        AT2=(SQRT(PI))*(1.0-ERF)
        AT3=ZET3**2
        AT4=EXP(-AT3)/ZET3
        AT5=(1.0/(2.0*AT3))-1.0
        ATN=AT1*(AT2+(AT4*AT5))
        TYPE 1111,ATN
        HX=H*30.48
        EM=HX*SQRT(ATN)
        TYPE 1111,EM
        EM1=(1.0/EM)
        CALL ERFS(EM1,ERF)
        BN1=((1.0+ERF)*SQRT(PI))/(2.0*EM)
        BN2=(1.0/(EM**2))
        BN3=((EXP(-BN2)/BN1)**2)*2.0
        BN4=((EM**2)*BN3)/(4.0*PI)
        TYPE 1111,BN4
        SXN3=SIND(THE0)/COSD(THE0)
        SXN4=SXN3/((SQRT(2.0))*SIGW)
        CALL ERFS(SXN4;ERF)
        SXN5=ERF
        SXN6=(1.0-SIND(THE0))/COSD(THE0)
        SXN7=SXN6/((SQRT(2.0))*SIGW)
        CALL ERFS(SXN7,ERF)
        BN5=(SXN5+ERF)
        BN6=BN5*BN4
        FO1=SXN3+(CY1*COS4)
        FO2=FO1/((SQRT(2.0))*SIGW)
        CALL ERFS(FO2,ERF)
        FO3=(1.0+ERF)/2.0
        FO4=((SIGW**2)+(FO1**2))*(COSD(THE0))**2
        FO5=(FO4*FO3)
        FO6=((FO1)**2)/(2.0*(SIGW**2))
        FO7=EXP(-FO6)
        FO8=(((COSD(THE0))**2)*FO7*SIGW)/SQRT(2.0*PI)
        FO9=(FO5+FO8)*BN6
        TYPE 3331,FO9
3331    FORMAT(1H , E10.4)
        TB=(CB*COSD(THE0))/CS
        TB1=ACOS(TB)
        TB2=(TB1*180.0)/PI
        IF(BT.EQ.3.0) GO TO 9999
        IF(BT.EQ.5.0) GO TO 9999
        TYPE 9996
9996    FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998    CALL BT3(TB2,PBL)
        GO TO 9997
9999    CALL BT5(TB2,PBL)
9997    TYPE 9995,TB2,PBL
9995    FORMAT(1H ,2E10.4)
        S1=(2.0*CS)/COSD(THE0)
        S2=ACOS((CMAX*COSD(THE0))/CS)
        S3=(((THE0*PI)/180.0)-S2)/GS
        S4=ACOS((CMIN*COSD(THE0))/CS)
        S5=(S4-S2)/G1
        S6=ACOS((CB*COSD(THE0))/CS)
        S7=(S4-S6)/GB
        S8=(S3+S5+S7)*S1
```

```
      TYPE 3332,S8
3332  FORMAT(1H , E10.4)
      FO10=-((2.0*RBL)+(2.0*A*S8))/10.0
      FO11=(10.0**FO10)
      FO12=FO9*FO11
      PQ1=(8.0*(AXN**2)*COSD(THE0))/(4.0*SIND(THE0)*B11*ABS(P15))
      PRSQ=PQ1*FO12
      TL=10.0*ALOG10(PRSQ)
      RETURN
      END
```

```
.TYPE BO.F4
      SUBROUTINE BOUN(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PPSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
      1 G1
      X1=(VAX*(T-(H/C1)))+XIA
      X2=(VS*T)+XIS
      X3=(VAY*(T-(H/C1)))+YIA
      X4=((X1-X2)**2)+X3**2
      X5=SQRT(X4)
      R1=(CMIN/CB)**2
      R2=SQRT(1.0-R1)
      R3=SQRT(1.0-(CS/CB)**2)
      R4=(R2-R3)/GS
      R5=R2/GB
      R6=(R4+R5)*2.0*CB
      IF(X5.GE.R6) GO TO 100
      TYPE 1
1     FORMAT(1H ,'THERE IS A SINGLE BOUNCE'/)
      THE8=ACOS(CS/CB)
      THE=(180.0*THE8)/3.14159265
      ZN=.01
11    DO 10 I=1,9000,1
      Z2=FLOAT(I)
      Z=THE+((Z2-1.0)/100.0)+.01
      IF(Z.GT.90.0) GO TO 400
      TH=CS/(CMIN*COSD(Z))
      TH1=ALOG(TH+SQRT((TH**2)-1.0))
      TH2=1.0/COSD(Z)
      TH3=ALOG(TH2+SQRT((TH2**2)-1.0))
      TH4=CS/(CB*COSD(Z))
      TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
      TH6=(2.0*(TH1-TH3))/GS
      TH7=(2.0*(TH5-TH1))/GB
      TH8=TH6-TH7
      X1=(VAX*(T-(H/C1)))+XIA
      X2=(VS*(T+TH8))+XIS
      X3=(VAY*(T-(H/C1)))+YIA
      X4=((X1-X2)**2)+X3**2
      X5=SQRT(X4)
      B1=((CMIN/CS)**2)*((COSD(Z))**2)
      B2=SQRT(1.0-B1)
      B3=(B2-SIND(Z))/GS
      B4=((CB/CS)**2)*((COSD(Z))**2)
      B5=SQRT(1.0-B4)
      B6=(B2-B5)/GB
      B7=((B3+B6)*2.0*CS)/(COSD(Z))
      B8=(B7-X5)/B7
      B9=ABS(B8)
      IF(B9.LE.ZN) GO TO 41
10    CONTINUE
      ZN=ZN+.01
      Q=.1
      IF(ZN.LE.Q) GO TO 11
400   TYPE 40
40    FORMAT(1H ,'NO SOLUTION TO SINGLE BOUNCE'/)
      GO TO 100
41    THE0=Z
      CALL BS(THE0)
      GO TO 42
```

```
100     TYPE 43
43      FORMAT(1H ,'TRYING A DOUBLE BOUNCE SOLUTION'/)
        THE9=ACOS(CS/CB)
        THE1=(180.0*THE9)/3.14159265
        ZN1=.01
12      DO 99 I=1,9000,1
        Z3=FLOAT(I)
        Z1=THE1+((Z3-1.0)/100.0)+.01
        IF(Z1.GT.90.0) GO TO 988
        THB=CS/(CMIN*COSD(Z1))
        THB1=ALOG(THB+SQRT((THB**2)-1.0))
        THB2=1.0/COSD(Z1)
        THB3=ALOG(THB2+SQRT((THB2**2)-1.0))
        THB4=CS/(CB*COSD(Z1))
        THB5=ALOG(THB4+SQRT((THB4**2)-1.0))
        THB6=(2.0*(THB1-THB3))/GS
        THB7=(2.0*(THB5-THB1))/GB
        THB8=THB6-THB7
        XB1=(VAX*(T-(H/C1)))+XIA
        XB2=(VS*(T+(2.0*THB8)))+XIS
        XB3=(VAY*(T-(H/C1)))+YIA
        XB4=((XB1-XB2)**2)+XB3**2
        XB5=SQRT(XB4)
        BB1=((CMIN/CS)**2)*((COSD(Z1))**2)
        BB2=SQRT(1.0-BB1)
        BB3=(BB2-SIND(Z1))/GS
        BB4=((CB/CS)**2)*((COSD(Z1))**2)
        BB5=SQRT(1.0-BB4)
        BB6=(BB2-BB5)/GB
        BB7=((BB3+BB6)*4.0*CS)/(COSD(Z1))
        BB8=(BB7-XB5)/BB7
        BB9=ABS(BB8)
        IF(BB9.LE.ZN1) GO TO 97
99      CONTINUE
        ZN1=ZN1+.01
        Q1=.1
        IF(ZN1.LE.Q1) GO TO 12
988     TYPE 98
98      FORMAT(1H ,'NO SOLUTION TO DOUBLE BOUNCE'/)
        GO TO 42
97      THE0=Z1
        CALL BD(THE0)
42      RETURN
        END
        SUBROUTINE BS(THE0)
        COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,ANN,BT
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DUP,DE,PHI,FFIO,A,MODE
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
        1 G1
        B1=((CMIN/CS)**2)*(COSD(THE0))**2
        B2=(SQRT(1.0-B1))-SIND(THE0)
        B3=((CB/CS)**2)*(COSD(THE0))**2
        B4=SQRT(1.0-B3)
        B5=(SQRT(1.0-B1))-B4
        B6=(B2/GS)+(B5/GB)
        B7=(2.0*CS*B6)/COSD(THE0)
        T1=CS/(CMIN*COSD(THE0))
        T2=ALOG(T1+SQRT((T1**2)-1.0))
        T3=1.0/COSD(THE0)
```

```
      T4=ALOG(T3+SQRT((T3**2)-1.0))
      T5=((T2-T4)*2.0)/GS
      T6=CS/(CB*COSD(THE0))
      T7=ALOG(T6+SQRT((T6**2)-1.0))
      T8=((T7-T2)*2.0)/GB
      T9=T5-T8
      DE=THE0
      TA=T+T9
      P1=B7*(SIND(THE0)/COSD(THE0))
      P2=((CMIN/CS)**2)*SIND(THE0)
      P3=((CMIN/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=((P2/P4)-1.0)/GS
      P6=(CB/CS)**2
      P7=P6*(COSD(THE0))**2
      P8=SQRT(1.0-P7)
      P9=P6/P8
      P10=(((CMIN/CS)**2)/P4)-P9
      P11=((SIND(THE0))*P10)/GB
      P12=((P5+P11)*2.0*CS)+P1
      X1=(VAX*(T-(H/C1)))+XIA
      X2=(VS*(T+T9))+XIS
      X3=(X1-X2)**2
      X4=(VAY*((T-(H/C1))))+YIA
      X5=(X1-X2)/SQRT(X3+(X4**2))
      X6=ACOS(X5)
      PHI=(180.0*X6)/PI
      CO1=(VS*(T+T9))+XIS
      CO2=(VAX*(T-(H/C1)))+XIA
      CO3=VAX*(CO1-CO2)
      CO4=-((VAY*(T-(H/C1)))+YIA)
      CO5=CO3+(VAY*CO4)
      CO6=SQRT(VAX**2+VAY**2)
      CO7=ABS(CO6)
      CO8=(CO1-CO2)**2
      CO9=CO8+(CO4**2)+((2.0*DB)-D)**2
      CO10=CO7*SQRT(CO9)
      CO11=CO5/CO10
      DO1=(CO6**2)/(C2**2)
      DO2=((2.0*CO7)*CO11)/C2
      DO3=1.0+DO1-DO2
      DO4=SQRT(DO3)
      DOP1=1.0/DO4
      DO5=(VS/C2)**2
      DO6=((SQRT(CO8))*2.0*VS)/C2
      DO7=1.0+DO5-(DO6/SQRT(CO9))
      DOP=(DOP1)*SQRT(DO7)
      MODE='BS'
      TDIF=T+T9-TO-(RO/C2)-(H/C1)
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT')
9998  CALL BT3(TB2,RBL)
      GO TO 9997
```

```
9999    CALL BT5(TB2,RBL)
9997    TYPE 9995,TB2,RBL
9995    FORMAT(1H ,2E10.3)
        S1=(2.0*CS)/COSD(THE0)
        S2=(1.0/GS)+(1.0/GB)
        S3=ASIN((CMIN*COSD(THE0))/CS)
        S4=ASIN((CB*COSD(THE0))/CS)
        S5=((PI/2.0)-((THE0*PI)/180.0))/GS
        S6=((-S2*S3)+(S4/GB)+S5)*S1
        QZ1=-((A*S6)+RBL)/10.0
        QZ2=(10.0**QZ1)
        PQ1=8.0*(AXN**2)*SIND(THE0)*COSD(THE0)*QZ2
        PRSQ=PQ1/(B7*ABS(P12))
        TL=(10.0*ALOG10(PRSQ))
        RETURN
        END
        SUBROUTINE BD(THE0)
        COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
        1 G1
        B1=((CMIN/CS)**2)*(COSD(THE0))**2
        B2=(SQRT(1.0-B1))-SIND(THE0)
        B3=((CB/CS)**2)*(COSD(THE0))**2
        B4=SQRT(1.0-B3)
        B5=(SQRT(1.0-B1))-B4
        B6=(B2/GS)+(B5/GB)
        B7=(2.0*CS*B6)/COSD(THE0)
        T1=CS/(CMIN*COSD(THE0))
        T2=ALOG(T1+SQRT((T1**2)-1.0))
        T3=1.0/COSD(THE0)
        T4=ALOG(T3+SQRT((T3**2)-1.0))
        T5=((T2-T4)*2.0)/GS
        T6=CS/(CB*COSD(THE0))
        T7=ALOG(T6+SQRT((T6**2)-1.0))
        T8=((T7-T2)*2.0)/GB
        T9=T5-T8
        DE=THE0
        TA=T+(2.0*T9)
        TDIF=T+(2.0*T9)-TO-(H/C1)-(RO/C2)
        P1=B7*(SIND(THE0)/COSD(THE0))
        P2=((CMIN/CS)**2)*SIND(THE0)
        P3=((CMIN/CS)**2)*(COSD(THE0))**2
        P4=SQRT(1.0-P3)
        P5=((P2/P4)-1.0)/GS
        P6=(CB/CS)**2
        P7=P6*(COSD(THE0))**2
        P8=SQRT(1.0-P7)
        P9=P6/P8
        P10=(((CMIN/CS)**2)/P4)-P9
        P11=((SIND(THE0))*P10)/GB
        P12=((P5+P11)*2.0*CS)+P1
        X1=(VAX*(T-(H/C1)))+XIA
        X2=(VS*(T+(2.0*T9)))+XIS
        X3=(X1-X2)**2
        X4=(VAY*((T-(H/C1))))+YIA
        X5=(X1-X2)/SQRT(X3+(X4**2))
        X6=ACOS(X5)
        PHI=(180.0*X6)/PI
```

```
      CO1=(VS*(T+(2.0*T9)))+XIS
      CO2=(VAX*(T-(H/C1)))+XIA
      CO3=VAX*(CO1-CO2)
      CO4=-((VAY*(T-(H/C1)))+YIA)
      CO5=CO3+(VAY*CO4)
      CO6=SQRT(VAX**2+VAY**2)
      CO7=ABS(CO6)
      CO8=(CO1-CO2)**2
      CO9=CO8+(CO4**2)+((4.0*DB)-D)**2
      CO10=CO7*SQRT(CO9)
      CO11=CO5/CO10
      DO1=(CO6**2)/(C2**2)
      DO2=((2.0*CO7)*CO11)/C2
      DO3=1.0+DO1-DO2
      DO4=SQRT(DO3)
      DOP1=1.0/DO4
      DO5=(VS/C2)**2
      DO6=((SQRT(CO8))*2.0*VS)/C2
      DO7=1.0+DO5-(DO6/SQRT(CO9))
      DOP=(DOP1)*SQRT(DO7)
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,RBL)
      GO TO 9997
9999  CALL BT5(TB2,RBL)
9997  TYPE 9995,TB2,RBL
9995  FORMAT(1H ,2E10.3)
      S1=(2.0*CS)/COSD(THE0)
      S2=(1.0/GS)+(1.0/GB)
      S3=ASIN((CMIN*COSD(THE0))/CS)
      S4=ASIN((CB*COSD(THE0))/CS)
      S5=((PI/2.0)-((THE0*PI)/180.0))/GS
      S6=((-S2*S3)+(S4/GB)+S5)*S1
      QZZ1=-((2.0*A*S6)+(2.0*PBL))/10.0
      QZZ2=(10.0**QZZ1)
      MODE='BD'
      PQ1=8.0*(AXN**2)*SIND(THE0)*COSD(THE0)*QZZ2
      PRSO=PQ1/((B7*ABS(P12))*4.0)
      TL=(10.0*ALOG10(PRSO))
      RETURN
      END
```

```
.TYPE SBO.F4
        SUBROUTINE SBOUN(THEO)
        COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSQ,A,MODE
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
        1 G1
        X1=(VAX*(T-(H/C1)))+XIA
        X2=(VS*T)+XIS
        X3=(VAY*(T-(H/C1)))+YIA
        X4=((X1-X2)**2)+X3**2
        X5=SQRT(X4)
        R1=SQRT(1.0-((CS/CB)**2))
        R2=SQRT(1.0-((CMAX/CB)**2))
        R3=(R1-R2)/GS
        R4=SQRT(1.0-((CMIN/CB)**2))
        R5=(R4-R2)/G1
        R6=R4/GB
        R7=(R3+R5+R6)*2.0*CB
        IF(X5.GE.R7) GO TO 100
        TYPE 1
1       FORMAT(1H ,'SURFACE DUCT WITH SINGLE BOUNCE SOLUTION'/)
        THE8=ACOS(CS/CB)
        THE=(180.0*THE8)/PI
        ZN=.01
11      DO 10 I=1,9000,1
        Z2=FLOAT(I)
        Z=THE+((Z2-1.0)/100.0)+.01
        IF(Z.GT.90.0) GO TO 400
        TH=ALOG((1.0/COSD(Z))+SQRT(((1.0/COSD(Z))**2)-1.0))
        TH1=CS/(CMAX*COSD(Z))
        TH2=ALOG(TH1+SQRT((TH1**2)-1.0))
        TH3=((TH-TH2)*2.0)/GS
        TH4=CS/(CMIN*COSD(Z))
        TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
        TH6=((TH5-TH2)*2.0)/G1
        TH7=CS/(CB*COSD(Z))
        TH8=ALOG(TH7+SQRT((TH7**2)-1.0))
        TH9=((TH5-TH8)*2.0)/GB
        TH10=TH3+TH6+TH9
        X1=(VAX*(T-(H/C1)))+XIA
        X2=(VS*(T+TH10))+XIS
        X3=(VAY*(T-(H/C1)))+YIA
        X4=((X1-X2)**2)+X3**2
        X5=SQRT(X4)
        B1=(2.0*CS)/COSD(Z)
        B2=((CMAX/CS)**2)*(COSD(Z))**2
        B3=SQRT(1.0-B2)
        B4=(SIND(Z)-B3)/GS
        B5=((CMIN/CS)**2)*(COSD(Z))**2
        B6=SQRT(1.0-B5)
        B7=(B6-B3)/G1
        B8=((CB/CS)**2)*(COSD(Z))**2
        B9=SQRT(1.0-B8)
        B10=(B6-B9)/GB
        B11=(B4+B7+B10)*B1
        B12=(B11-X5)/B11
        B13=ABS(B12)
        IF(B13.LE.ZN) GO TO 41
10      CONTINUE
        ZN=ZN+.01
        Q=.1
```

```
      IF(ZN.LE.Q) GO TO 11
400   TYPE 40
40    FORMAT(1H ,'NO SOLUTION OF SURFACE DUCT WITH SINGLE BOUNCE'/)
      GO TO 100
41    THE0=Z
      CALL BSS(THE0)
      GO TO 42
100   TYPE 43
43    FORMAT(1H ,'TRYING A SURFACE DUCT WITH DOUBLE BOUNCE SOLUTION'/)
      THE9=ACOS(CS/CB)
      THE1=(180.0*THE9)/PI
      ZN1=.01
12    DO 99 I=1,9000,1
      Z3=FLOAT(I)
      Z1=THE1+((Z3-1.0)/100.0)+.01
      IF(Z1.GT.90.0) GO TO 988
      THB=ALOG((1.0/COSD(Z1))+SQRT(((1.0/COSD(Z1))**2)-1.0))
      THB1=CS/(CMAX*COSD(Z1))
      THB2=ALOG(THB1+SQRT((THB1**2)-1.0))
      THB3=((THB-THB2)*2.0)/GS
      THB4=CS/(CMIN*COSD(Z1))
      THB5=ALOG(THB4+SQRT((THB4**2)-1.0))
      THB6=((THB5-THB2)*2.0)/G1
      THB7=CS/(CB*COSD(Z1))
      THB8=ALOG(THB7+SQRT((THB7**2)-1.0))
      THB9=((THB5-THB8)*2.0)/GB
      THB10=THB3+THB6+THB9
      XB1=(VAX*(T-(H/C1)))+XIA
      XB2=(VS*(1+(2.0*THB10)))+X1S
      XB3=(VAY*(T-(H/C1)))+Y1A
      XB4=((XB1-XB2)**2)+XB3**2
      XB5=SQRT(XB4)
      BB1=(2.0*CS)/COSD(Z1)
      BB2=((CMAX/CS)**2)*(COSD(Z1))**2
      BB3=SQRT(1.0-BB2)
      BB4=(SIND(Z1)-BB3)/GS
      BB5=((CMIN/CS)**2)*(COSD(Z1))**2
      BB6=SQRT(1.0-BB5)
      BB7=(BB6-BB3)/G1
      BB8=((CB/CS)**2)*(COSD(Z1))**2
      BB9=SQRT(1.0-BB8)
      BB10=(BB6-BB9)/GB
      BB11=(BB4+BB7+BB10)*BB1*2.0
      BB12=(BB11-XB5)/BB11
      BB13=ABS(BB12)
      IF(BB13.LE.ZN1) GO TO 97
99    CONTINUE
      ZN1=ZN1+.01
      Q1=.1
      IF(ZN1.LE.Q1) GO TO 12
988   TYPE 98
98    FORMAT(1H ,'NO SOLUTION OF SURFACE DUCT WITH DOUBLE BOUNCE'/)
      GO TO 42
97    THE0=Z1
      CALL BDS(THE0)
42    RETURN
      END
```

```
      SUBROUTINE BSS(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PPSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      B1=(2.0*CS)/COSD(THE0)
      B2=((CMAX/CS)**2)*(COSD(THE0))**2
      B3=SQRT(1.0-B2)
      B4=(SIND(THE0)-B3)/GS
      B5=((CMIN/CS)**2)*(COSD(THE0))**2
      B6=SQRT(1.0-B5)
      B7=(B6-B3)/G1
      B8=((CB/CS)**2)*(COSD(THE0))**2
      B9=SQRT(1.0-B8)
      B10=(B6-B9)/GB
      B11=(B4+B7+B10)*B1
      TH=ALOG((1.0/COSD(THE0))+SQRT(((1.0/COSD(THE0))**2)-1.0))
      TH1=CS/(CMAX*COSD(THE0))
      TH2=ALOG(TH1+SQRT((TH1**2)-1.0))
      TH3=((TH-TH2)*2.0)/GS
      TH4=CS/(CMIN*COSD(THE0))
      TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
      TH6=((TH5-TH2)*2.0)/G1
      TH7=CS/(CB*COSD(THE0))
      TH8=ALOG(TH7+SQRT((TH7**2)-1.0))
      TH9=((TH5-TH8)*2.0)/GB
      TH10=TH3+TH6+TH9
      DE=THE0
      TA=T+TH10
      P1=B11*(SIND(THE0)/COSD(THE0))
      P2=((CMAX/CS)**2)*SIND(THE0)
      P3=((CMAX/CS)**2)*(COSD(THE0))**2
      P4=SQRT(1.0-P3)
      P5=(1.0-(P2/P4))/GS
      P6=(CMIN/CS)**2
      P7=SQRT(1.0-(P6*((COSD(THE0))**2)))
      P8=P6/P7
      P9=(CMAX/CS)**2
      P10=P9/P4
      P11=((P8-P10)*SIND(THE0))/G1
      P12=(CB/CS)**2
      P13=SQRT(1.0-(P12*((COSD(THE0))**2)))
      P14=P12/P13
      P16=((P8-P14)*SIND(THE0))/GB
      P15=((P5+P11+P16)*2.0*CS)+P1
      TYPE 3334,B11,P15
3334  FORMAT(1H ,2E10.4)
      TYPE 3335,G1,GS,GB
3335  FORMAT(1H ,3F10.3)
      TYPE 3336,C1,CS,CMAX,CMIN,CB
3336  FORMAT(1H ,5F10.3)
      X1=(VAX*(T-(H/C1)))+XIA
      X2=(VS*(T+TH10))+XIS
      X3=(X1-X2)**2
      X4=(VAY*(T-(H/C1)))+YIA
      X5=(X1-X2)/SQRT(X3+(X4**2))
      X6=ACOS(X5)
      PHI=(180.0*X6)/PI
      CO1=(VS*(T+TH10))+XIS
      CO2=(VAX*(T-(H/C1)))+XIA
```

```
      CO3=VAX*(CO1-CO2)
      CO4=-(VAY*(T-(H/C1)))+YIA
      CO5=CO3+(VAY*CO4)
      CO6=SQRT(VAX**2+VAY**2)
      CO7=ABS(CO6)
      CO8=(CO1-CO2)**2
      CO9=CO8+(CO4**2)+((2.0*DB)-D)**2
      CO10=CO7*SQRT(CO9)
      CO11=CO5/CO10
      DO1=(CO6**2)/(C2**2)
      DO2=((2.0*CO7)*CO11)/C2
      DO3=1.0+DO1-DO2
      DO4=SQRT(DO3)
      DOP1=1.0/DO4
      DO5=(VS/C2)**2
      DO6=((SQRT(CO8))*2.0*VS)/C2
      DO7=1.0+DO5-(DO6/SQRT(CO9))
      DOP=(DOP1)*SQRT(DO7)
      MODE='BSS'
      TDIF=T+TH10-TD-(RD/C2)-(H/C1)
      TB=(CB*COSD(THE0))/CS
      TB1=ACOS(TB)
      TB2=(TB1*180.0)/PI
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,RBL)
      GO TO 9997
9999  CALL BT5(TB2,RBL)
9997  TYPE 9995,TB2,RBL
9995  FORMAT(1H ,2E10.3)
      S1=(2.0*CS)/COSD(THE0)
      S2=ACOS((CMAX*COSD(THE0))/CS)
      S3=(((THE0*PI)/180.0)-S2)/GS
      S4=ACOS((CMIN*COSD(THE0))/CS)
      S5=(S4-S2)/G1
      S6=ACOS((CB*COSD(THE0))/CS)
      S7=(S4-S6)/GB
      S8=(S3+S5+S7)*S1
      QZ1=-((A*S8)+RBL)/10.0
      QZ2=(10.0**QZ1)
      PO1=8.0*(AXN**2)*SIND(THE0)*COSD(THE0)*QZ2
      PRSO=PO1/(B11*ABS(P15))
      TL=(10.0*ALOG10(PRSO))
      RETURN
      END
      SUBROUTINE BDS(THE0)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,PI
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,PRSO,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RD,TD,PI,
     1 G1
      B1=(2.0*CS)/COSD(THE0)
      B2=((CMAX/CS)**2)*(COSD(THE0))**2
      B3=SQRT(1.0-B2)
      B4=(SIND(THE0)-B3)/GS
      B5=((CMIN/CS)**2)*(COSD(THE0))**2
      B6=SQRT(1.0-B5)
      B7=(B6-B3)/G1
      B8=((CB/CS)**2)*(COSD(THE0))**2
      B9=SQRT(1.0-B8)
```

```
B10=(B6-B9)/GB
B11=(B4+B7+B10)*B1
TH=ALOG((1.0/COSD(THE0))+SQRT(((1.0/COSD(THE0))**2)-1.0))
TH1=CS/(CMAX*COSD(THE0))
TH2=ALOG(TH1+SQRT((TH1**2)-1.0))
TH3=((TH-TH2)*2.0)/GS
TH4=CS/(CMIN*COSD(THE0))
TH5=ALOG(TH4+SQRT((TH4**2)-1.0))
TH6=((TH5-TH2)*2.0)/G1
TH7=CS/(CB*COSD(THE0))
TH8=ALOG(TH7+SQRT((TH7**2)-1.0))
TH9=((TH5-TH8)*2.0)/GB
TH10=TH3+TH6+TH9
DE=THE0
TA=T+(2.0*TH10)
P1=B11*(SIND(THE0)/COSD(THE0))
P2=((CMAX/CS)**2)*SIND(THE0)
P3=((CMAX/CS)**2)*(COSD(THE0))**2
P4=SQRT(1.0-P3)
P5=(1.0-(P2/P4))/GS
P6=(CMIN/CS)**2
P7=SQRT(1.0-(P6*((COSD(THE0))**2)))
P8=P6/P7
P9=(CMAX/CS)**2
P10=P9/P4
P11=((P8-P10)*SIND(THE0))/G1
P12=(CB/CS)**2
P13=SQRT(1.0-(P12*((COSD(THE0))**2)))
P14=P12/P13
P16=((P8-P14)*SIND(THE0))/GB
P15=((P5+P11+P16)*2.0*CS)+P1
X1=(VAX*(T-(H/C1)))+XIA
X2=(VS*(T+(2.0*TH10)))+XIS
X3=(X1-X2)**2
X4=(VAY*(T-(H/C1)))+YIA
X5=(X1-X2)/SQRT(X3+(X4**2))
X6=ACOS(X5)
PHI=(180.0*X6)/PI
CO1=(VS*(T+(2.0*TH10)))+XIS
CO2=(VAX*(T-(H/C1)))+XIA
CO3=VAX*(CO1-CO2)
CO4=-(VAY*(T-(H/C1)))+YIA
CO5=CO3+(VAY*CO4)
CO6=SQRT(VAX**2+VAY**2)
CO7=ABS(CO6)
CO8=(CO1-CO2)**2
CO9=CO8+(CO4**2)+((4.0*DB)-D)**2
CO10=CO7*SQRT(CO9)
CO11=CO5/CO10
DO1=(CO6**2)/(C2**2)
DO2=((2.0*CO7)*CO11)/C2
DO3=1.0+DO1-DO2
DO4=SQRT(DO3)
DOP1=1.0/DO4
DO5=(VS/C2)**2
DO6=((SQRT(CO8))*2.0*VS)/C2
DO7=1.0+DO5-(DO6/SQRT(CO9))
DOP=(DOP1)*SQRT(DO7)
TB=(CB*COSD(THE0))/CS
TB1=ACOS(TB)
TB2=(TB1*180.0)/PI
```

```
      IF(BT.EQ.3.0) GO TO 9998
      IF(BT.EQ.5.0) GO TO 9999
      TYPE 9996
9996  FORMAT(1H ,'DID NOT HAVE A VALUE FOR BT'/)
9998  CALL BT3(TB2,RBL)
      GO TO 9997
9999  CALL BT5(TB2,RBL)
9997  TYPE 9995,TB2,RBL
9995  FORMAT(1H ,2E10.3)
      S1=(2.0*CS)/COSD(THE0)
      S2=ACOS((CMAX*COSD(THE0))/CS)
      S3=(((THE0*PI)/180.0)-S2)/GS
      S4=ACOS((CMIN*COSD(THE0))/CS)
      S5=(S4-S2)/G1
      S6=ACOS((CB*COSD(THE0))/CS)
      S7=(S4-S6)/GB
      S8=(S3+S5+S7)*S1
      QZZ1=-((2.0*A*S8)+(2.0*RBL))/10.0
      QZZ2=(10.0**QZZ1)
      PQ1=8.0*(AXN**2)*SIND(THE0)*COSD(THE0)*QZZ2
      PRSQ=PQ1/(((B11*ABS(P15)))*4.0)
      TL=(10.0*ALOG10(PRSQ))
      MODE='BDS'
      TDIF=T+(2.0*TH10)-TO-(RO/C2)-(H/C1)
      RETURN
      END
```

```
TYPE BT3.F4
        SUBROUTINE BT3(TB2,RBL)
        COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
        COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOP,DE,PHI,FRSQ,A,MODE
        COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
        1 G1
        IF(F.LE.300.0) GO TO 1
        IF(F.GT.300.0.AND.F.LT.750.0) GO TO 2
        IF(F.GE.750.0.AND.F.LE.1500.0) GO TO 3
        IF(F.GT.1500.0.AND.F.LE.2700.0) GO TO 4
        IF(F.GT.2700.0.AND.F.LE.5000.0) GO TO 5
        IF(F.GT.5000.0) GO TO 6
        TYPE 556
556     FORMAT(1H ,'OUTSIDE FREQ. RANGE OF PROGRAM'/)
1       IF(TB2.LE.11.0) GO TO 8
        GO TO 200
8       RBL=0.0
        GO TO 100
200     IF(TB2.GT.11.0.AND.TB2.LT.50.0) GO TO 9
        GO TO 300
9       RBL=((10.0*TB2)/39.0)-2.82
        GO TO 100
300     IF(TB2.GE.50.0) GO TO 10
        GO TO 500
10      RBL=10.0
        GO TO 100
2       IF(TB2.LE.13.0) GO TO 11
        GO TO 400
11      RBL=3.0
        GO TO 100
400     IF(TB2.GT.13.0.AND.TB2.LT.20.0) GO TO 12
        GO TO 600
12      RBL=((2.3*TB2)/7.0)-1.27
        GO TO 100
600     IF(TB2.GE.20.0.AND.TB2.LE.35.0) GO TO 13
        GO TO 700
13      RBL=((3.4*TB2)/15.0)-.77
        GO TO 100
700     IF(TB2.GT.35.0.AND.TB2.LT.52.0) GO TO 14
        GO TO 701
14      RBL=((1.6*TB2)/10.0)+3.1
        GO TO 100
701     IF(TB2.GE.52.0) GO TO 15
        GO TO 500
15      RBL=11.0
        GO TO 100
3       IF(TB2.LE.15.0) GO TO 16
        GO TO 702
16      RBL=3.0
        GO TO 100
702     IF(TB2.GT.15.0.AND.TB2.LT.45.0) GO TO 17
        GO TO 703
17      RBL=((8.0*TB2)/30.0)-1.0
        GO TO 100
703     IF(TB2.GE.45.0) GO TO 18
        GO TO 500
18      RBL=11.0
        GO TO 100
4       IF(TB2.LE.13.0) GO TO 19
        GO TO 704
```

```
19      RBL=3.0
        GO TO 100
704     IF(TB2.GT.13.0.AND.[illegible].T.42.0) GO TO 20
        GO TO 705
20      RBL=((8.0*TB2)/29.0)-.59
        GO TO 100
705     IF(TB2.GE.42.0) GO TO 21
        GO TO 500
21      RBL=11.0
        GO TO 100
5       IF(TB2.LE.2.5) GO TO 22
        GO TO 706
22      RBL=7.0
        TYPE 1111
1111    FORMAT(1H ,'RBL HAS ERROR GREATER THAN 2 DB'/)
        GO TO 100
706     IF(TB2.GT.2.5.AND.TB2.LE.12.5) GO TO 23
        GO TO 707
23      RBL=((7.0*TB2)/10.0)+5.25
        TYPE 1111
        GO TO 100
707     IF(TB2.GT.12.5.AND.TB2.LT.32.5) GO TO 24
        GO TO 708
24      RBL=((1.5*TB2)/20.0)+13.06
        TYPE 1111
        GO TO 100
708     IF(TB2.GE.32.5) GO TO 25
        GO TO 500
25      RBL=15.5
        TYPE 1111
        GO TO 100
6       IF(TB2.LE.7.5) GO TO 26
        GO TO 709
26      RBL=8.0
        GO TO 100
709     IF(TB2.GT.7.5.AND.TB2.LT.15.0) GO TO 27
        GO TO 710
27      RBL=((5.0*TB2)/7.5)+3.0
        GO TO 100
710     IF(TB2.GE.15.0.AND.TB2.LT.25.0) GO TO 28
        GO TO 711
28      RBL=((3.0*TB2)/10.0)+8.5
        GO TO 100
711     IF(TB2.GE.25.0) GO TO 29
        GO TO 500
29      RBL=16.0
        GO TO 100
500     TYPE 555
555     FORMAT(1H ,'CANT FIND REFLECTION LOSS'/)
100     RETURN
        END
```

```
TYPE BT5.F4
      SUBROUTINE BT5(TB2,RBL)
      COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
      COMMON /OUT/ T,R,RP,TA,TDIF,TL,DOF,DE,PHI,PRSQ,A,MODE
      COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RO,TO,PI,
     1 G1
      IF(F.LE.300.0) GO TO 1
      IF(F.GT.300.0.AND.F.LT.750.0) GO TO 2
      IF(F.GE.750.0.AND.F.LE.1500.0) GO TO 3
      IF(F.GT.1500.0.AND.F.LE.2700.0) GO TO 4
      IF(F.GT.2700.0.AND.F.LE.5000.0) GO TO 5
      IF(F.GT.5000.0) GO TO 6
      TYPE 556
556   FORMAT(1H ,'OUTSIDE RANGE OF FREQS.'/)
      GO TO 100
1     IF(TB2.LE.5.0) GO TO 8
      GO TO 200
8     RBL=4.0
      GO TO 100
200   IF(TB2.GT.5.0.AND.TB2.LT.21.0) GO TO 9
      GO TO 300
9     RBL=((10.0*TB2)/16.0)+.88
      GO TO 100
300   IF(TB2.GE.21.0) GO TO 10
      GO TO 500
10    RBL=14.0
      GO TO 100
2     IF(TB2.LE.3.0) GO TO 11
      GO TO 400
11    RBL=8.0
      GO TO 100
400   IF(TB2.GT.3.0.AND.TB2.LT.22.0) GO TO 12
      GO TO 600
12    RBL=((8.0*TB2)/19.0)+6.74
      GO TO 100
600   IF(TB2.GE.22.0) GO TO 13
      GO TO 500
13    RBL=16.0
      GO TO 100
3     IF(TB2.LE.2.5) GO TO 16
      GO TO 702
16    RBL=9.0
      GO TO 100
702   IF(TB2.GT.2.5.AND.TB2.LT.17.5) GO TO 17
      GO TO 703
17    RBL=((9.0*TB2)/15.0 +7.5
      GO TO 100
703   IF(TB2.GE.17.5) GO TO 18
      GO TO 500
18    RBL=18.0
      GO TO 100
4     IF(TB2.LE.2.5) GO TO 19
      GO TO 704
19    RBL=9.0
      GO TO 100
704   IF(TB2.GT.2.5.AND.TB2.LT.20.0) GO TO 20
      GO TO 705
20    RBL=((9.0*TB2)/17.5)+7.71
      GO TO 100
705   IF(TB2.GE.20.0) GO TO 21
      GO TO 500
```

```
21      RBL=18.0
        GO TO 100
5       IF(TB2.LE.2.5) GO TO 22
        GO TO 706
22      RBL=11.0
        GO TO 100
706     IF(TB2.GT.2.5.AND.TB2.LT.12.5) GO TO 23
        GO TO 707
23      RBL=((7.1*TB2)/10.0)+9.23
        GO TO 100
707     IF(TB2.GE.12.5.AND.TB2.LT.22.5) GO TO 24
        GO TO 708
24      RBL=((1.9*TB2)/10.0)+15.73
        GO TO 100
708     IF(TB2.GE.22.5) GO TO 25
        GO TO 500
25      RBL=20.0
        GO TO 100
6       IF(TB2.LE.2.5) GO TO 26
        GO TO 709
26      RBL=10.0
        GO TO 100
709     IF(TB2.GT.2.5.AND.TB2.LT.10.0) GO TO 27
        GO TO 710
27      RBL=((6.0*TB2)/7.5)+8.0
        GO TO 100
710     IF(TB2.GE.10.0.AND.TB2.LT.20.0) GO TO 28
        GO TO 711
28      RBL=((3.0*TB2)/10.0)+13.0
        GO TO 100
711     IF(TB2.GE.20.0) GO TO 29
        GO TO 500
29      RBL=19.0
        GO TO 100
500     TYPE 555
555     FORMAT(1H ,'CANT FIND REFLECTION LOSS'/)
100     RETURN
        END
```



```
TYPE ERFS.F4
      SUBROUTINE ERFS(ARG,ERF)
      SARG=ARG/ABS(ARG)
      X=ABS(ARG)
      ER=1.0
      IF(X.GT.4.2) GO TO 200
      E=1.0+.0705230784*X+.0422820123*X*X+.0092705272*(X**3)+
     1 .0001520143*(X**4)+.0002765672*(X**5)+.0000430638*(X**6)
      ER=1.0-1.0/(E**16)
200   ERF=ER*SARG
      RETURN
      END
```